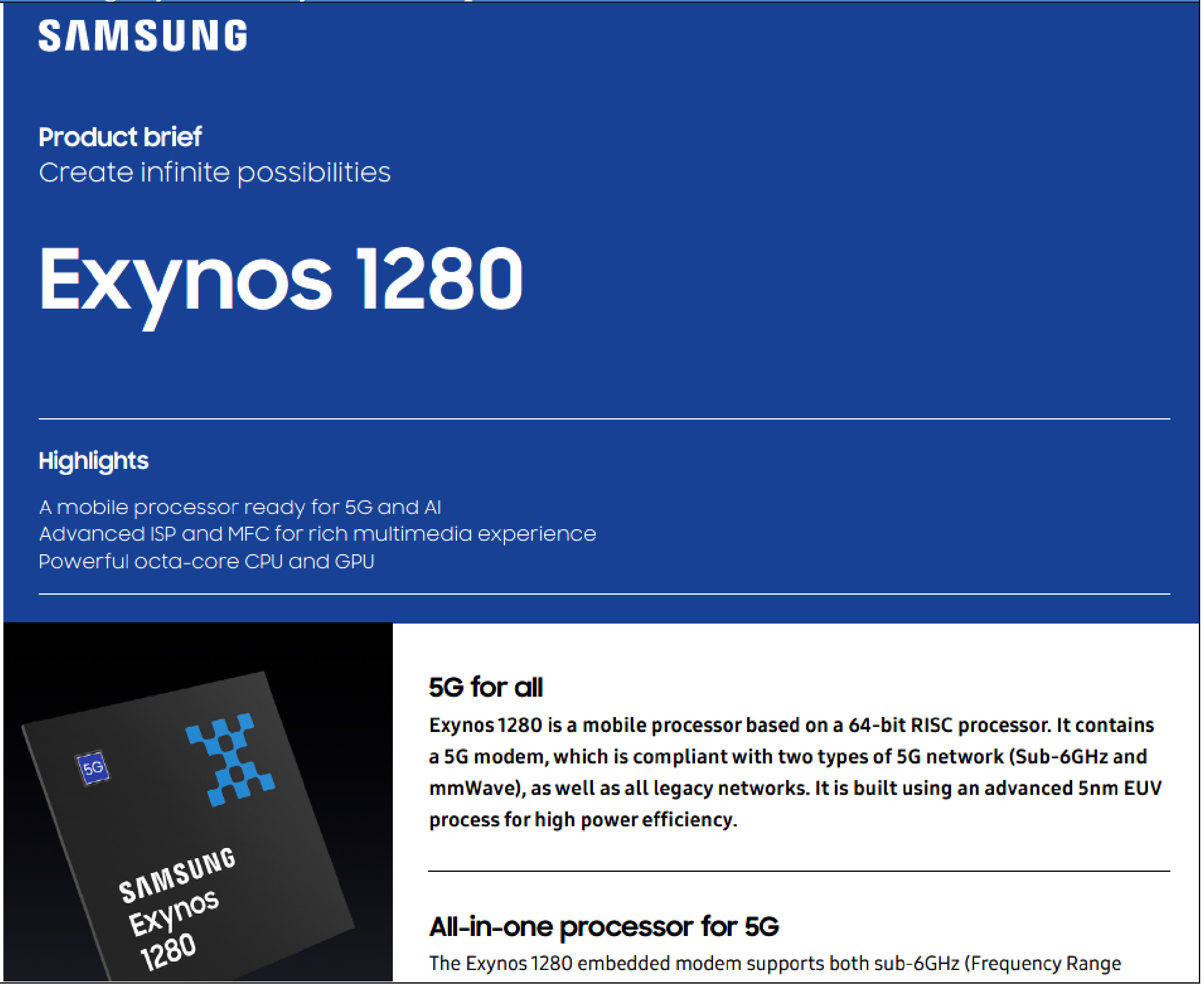
EXHIBIT 025

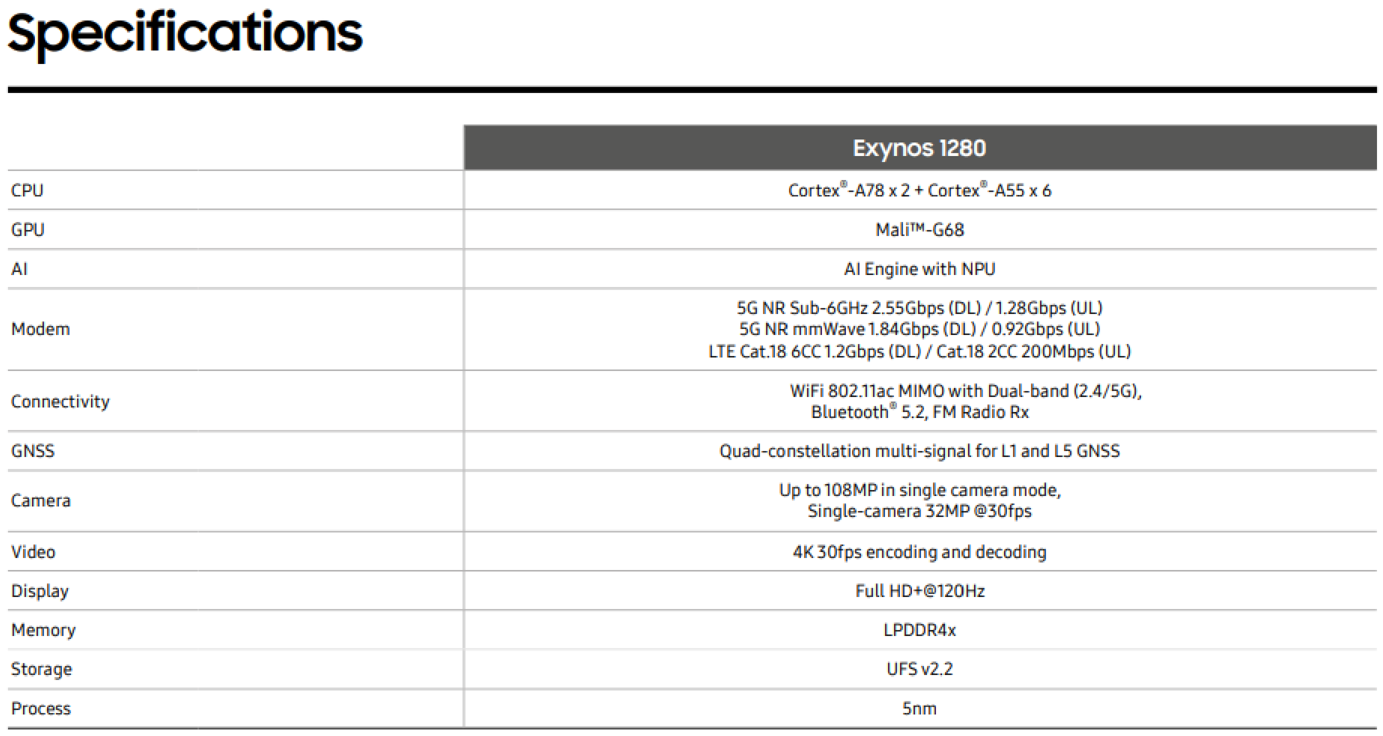
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
10. A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions, the method comprising the acts of:	Without conceding that the preamble of claim 10 of the '800 Patent is limiting, Samsung Electronics Co., Ltd.'s (hereinafter, "Samsung") Exynos 1280 system on chip (hereinafter, the "Exynos SoC") is an integrated circuit and performs a method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions), either literally or under the doctrine of equivalents.

[1] The Exynos SoC is charted as a representative product made used, sold, offered for sale, and/or imported by or on behalf of Samsung. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

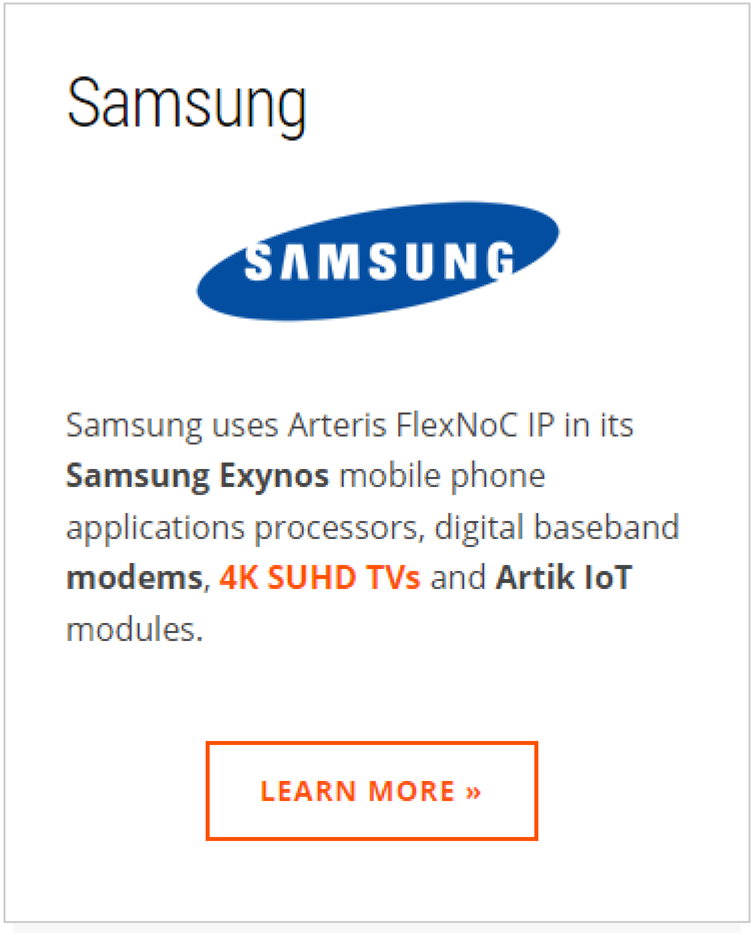
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 SAMSUNG **Product brief** Create infinite possibilities Exynos 1280 **Highlights** A mobile processor ready for 5G and AI Advanced ISP and MFC for rich multimedia experience Powerful octa-core CPU and GPU **5G for all** Exynos 1280 is a mobile processor based on a 64-bit RISC processor. It contains a 5G modem, which is compliant with two types of 5G network (Sub-6GHz and mmWave), as well as all legacy networks. It is built using an advanced 5nm EUV process for high power efficiency. **All-in-one processor for 5G** The Exynos 1280 embedded modem supports both sub-6GHz (Frequency Range

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	https://semiconductor.samsung.com/resources/brochure/Exynos1280.pdf The Exynos SoC comprises a plurality of processing modules, for example Arm Cortex-A78 core, Cortex-A55 core, Arm Mali-G68 GPU, and AI Engine with NPU:  https://semiconductor.samsung.com/resources/brochure/Exynos1280.pdf

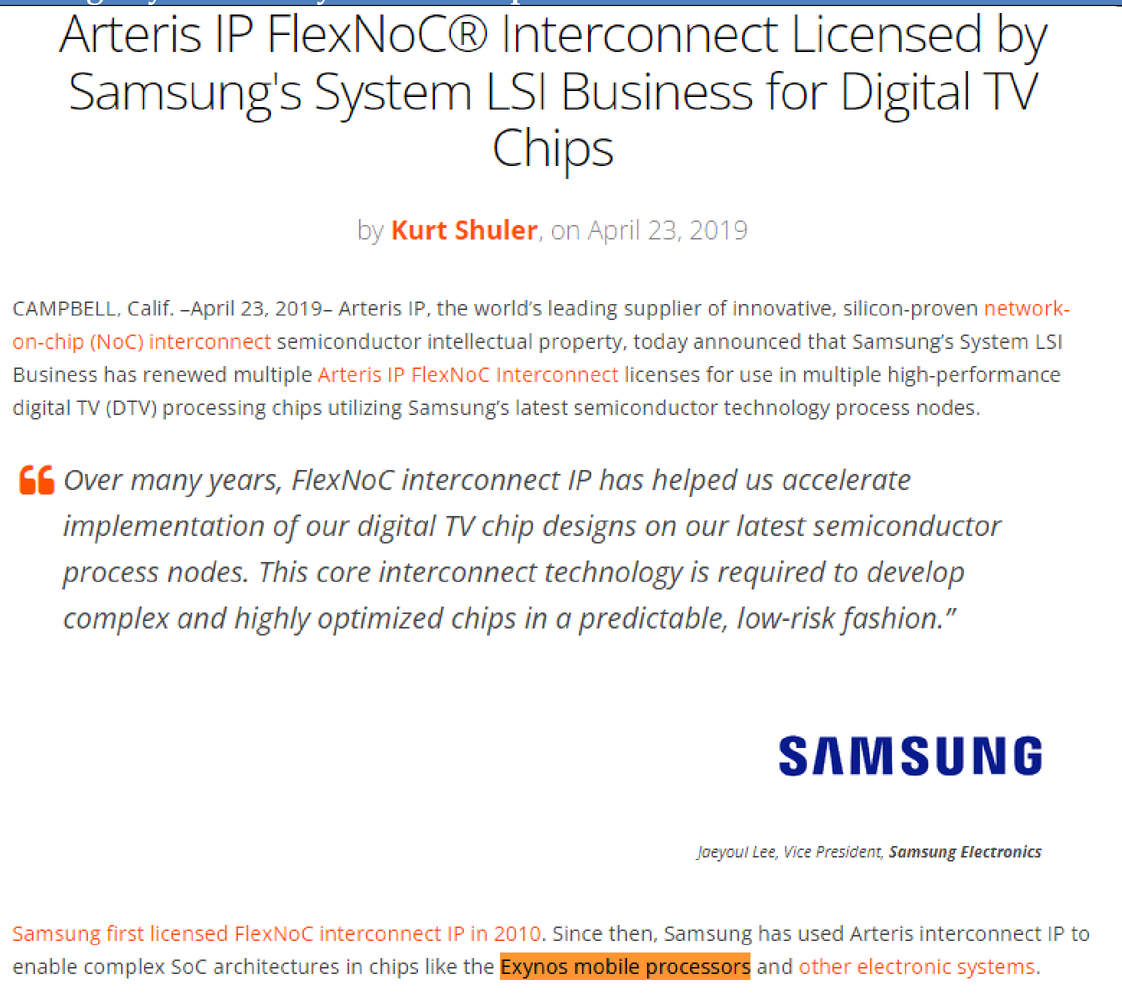
Specifications

	Exynos 1280
CPU	Cortex®-A78 x 2 + Cortex®-A55 x 6
GPU	Mali™-G68
AI	AI Engine with NPU
Modem	5G NR Sub-6GHz 2.55Gbps (DL) / 1.28Gbps (UL) 5G NR mmWave 1.84Gbps (DL) / 0.92Gbps (UL) LTE Cat.18 6CC 1.2Gbps (DL) / Cat.18 2CC 200Mbps (UL)
Connectivity	WiFi 802.11ac MIMO with Dual-band (2.4/5G), Bluetooth® 5.2, FM Radio Rx
GNSS	Quad-constellation multi-signal for L1 and L5 GNSS
Camera	Up to 108MP in single camera mode, Single-camera 32MP @30fps
Video	4K 30fps encoding and decoding
Display	Full HD+@120Hz
Memory	LPDDR4x
Storage	UFS v2.2
Process	5nm

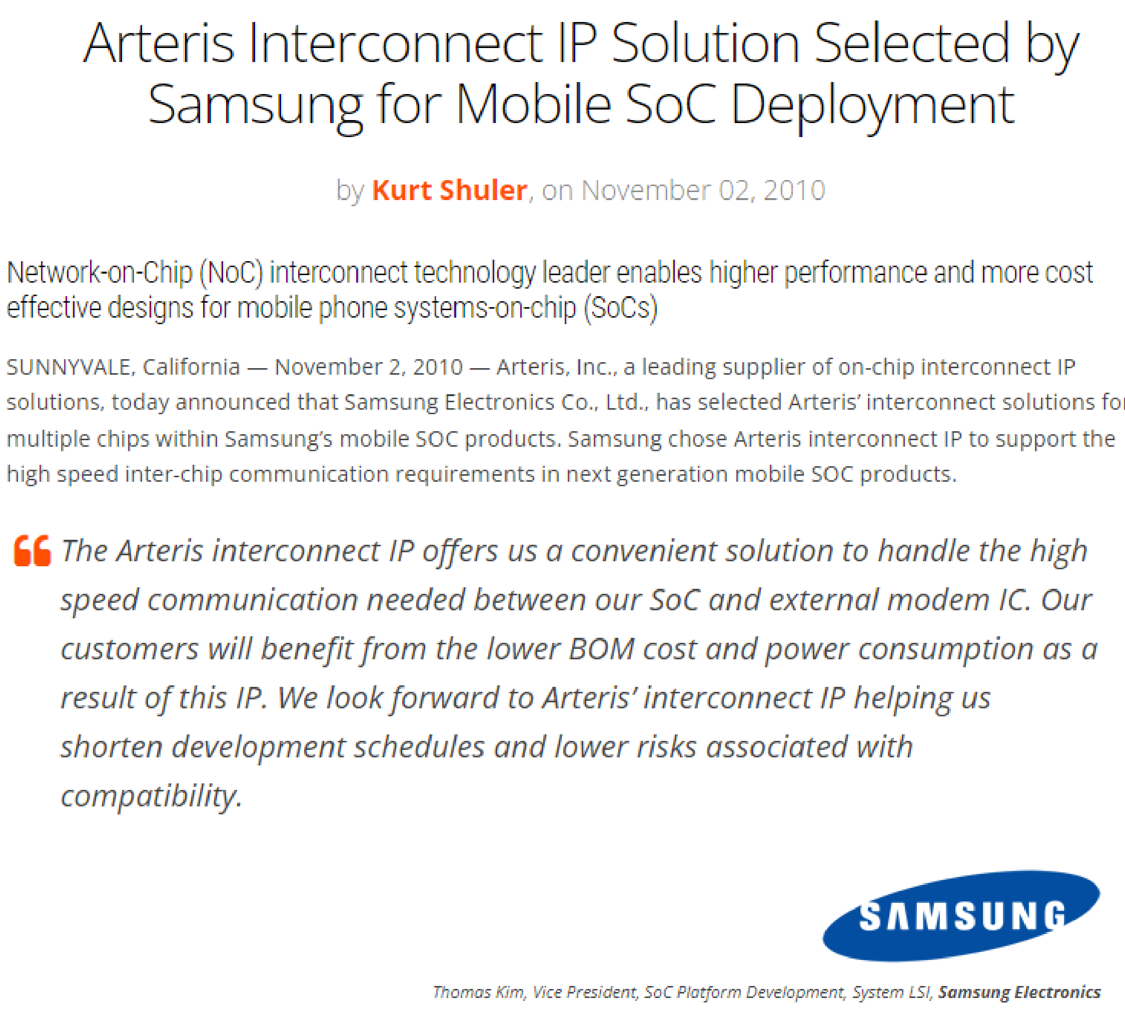
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	The Exynos SoC utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the "Arteris NoC") as an interconnect to connect the plurality of processing modules through interface units:  Samsung Samsung uses Arteris FlexNoC IP in its **Samsung Exynos** mobile phone applications processors, digital baseband **modems**, **4K SUHD TVs** and **Artik IoT** modules. LEARN MORE » https://web.archive.org/web/20210514110614/https://www.arteris.com/customers

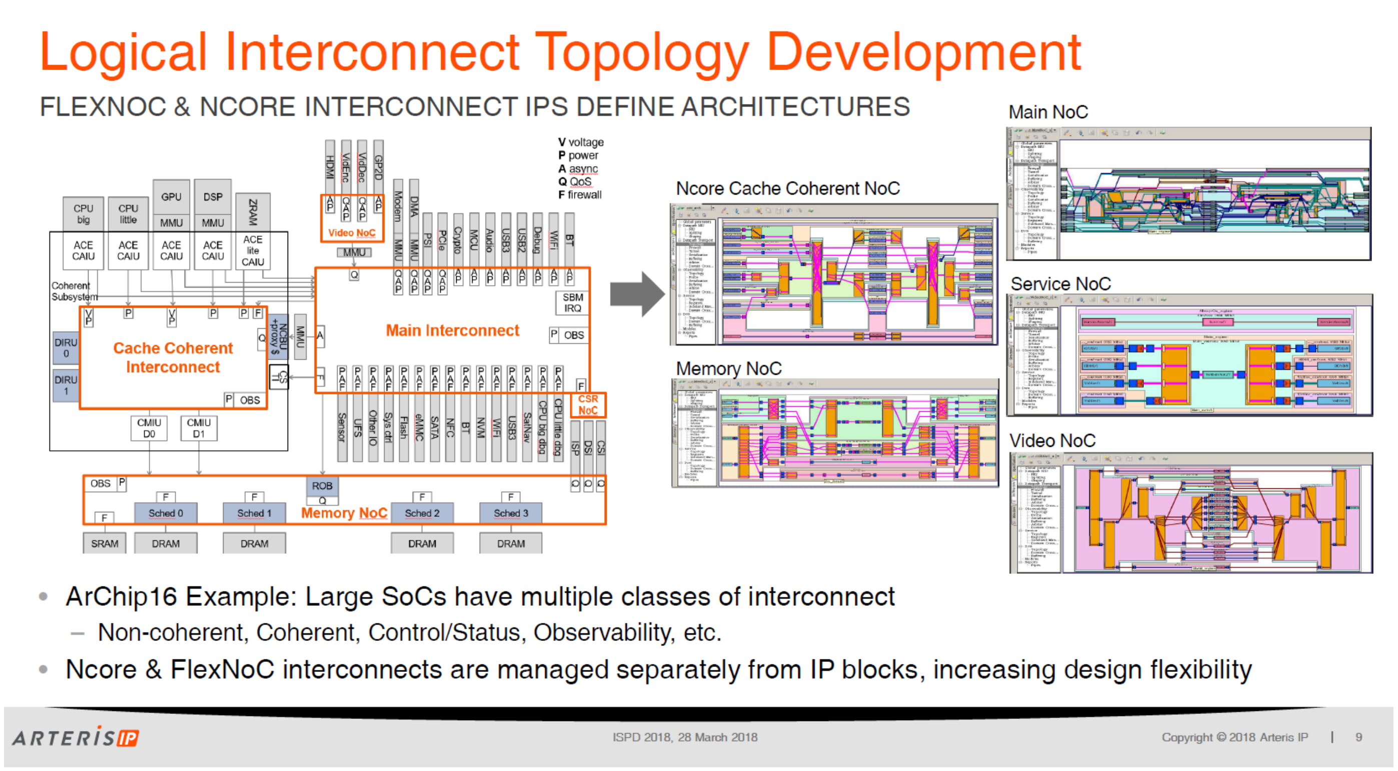
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Arteris IP FlexNoC® Interconnect Licensed by Samsung's System LSI Business for Digital TV Chips by **Kurt Shuler**, on April 23, 2019 CAMPBELL, Calif. –April 23, 2019– Arteris IP, the world's leading supplier of innovative, silicon-proven network-on-chip (NoC) interconnect semiconductor intellectual property, today announced that Samsung's System LSI Business has renewed multiple Arteris IP FlexNoC Interconnect licenses for use in multiple high-performance digital TV (DTV) processing chips utilizing Samsung's latest semiconductor technology process nodes. *Over many years, FlexNoC interconnect IP has helped us accelerate implementation of our digital TV chip designs on our latest semiconductor process nodes. This core interconnect technology is required to develop complex and highly optimized chips in a predictable, low-risk fashion."* SAMSUNG *Jaeyoul Lee, Vice President,* ***Samsung Electronics*** Samsung first licensed FlexNoC interconnect IP in 2010. Since then, Samsung has used Arteris interconnect IP to enable complex SoC architectures in chips like the Exynos mobile processors and other electronic systems. https://www.arteris.com/press-releases/samsung-lsi-dtv-arteris-ip-flexnoc

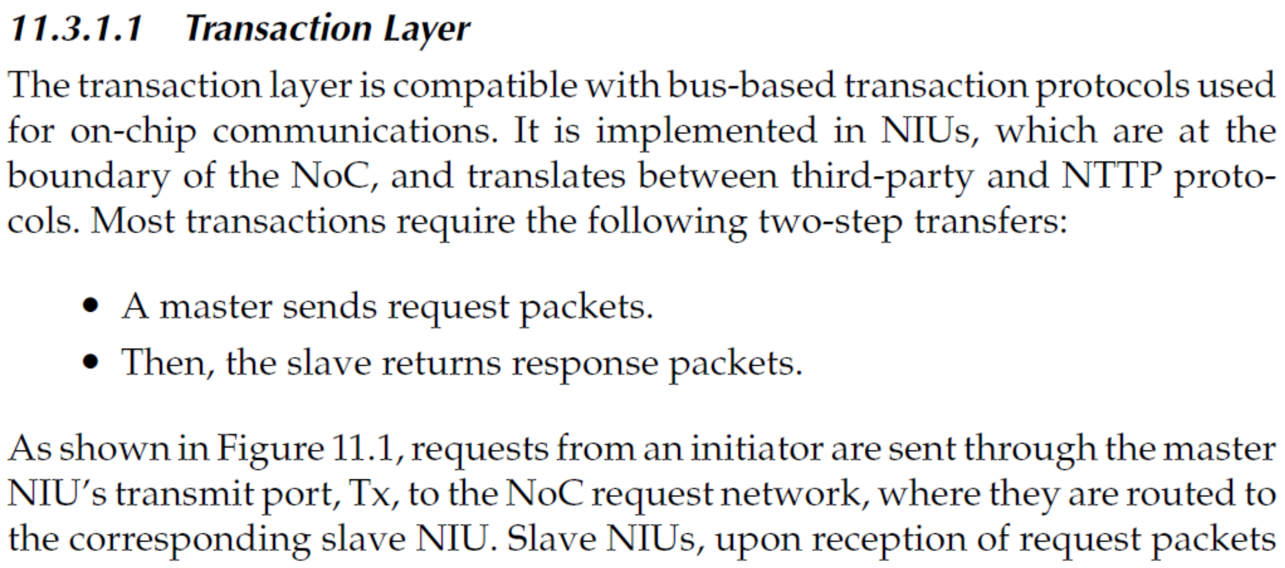
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

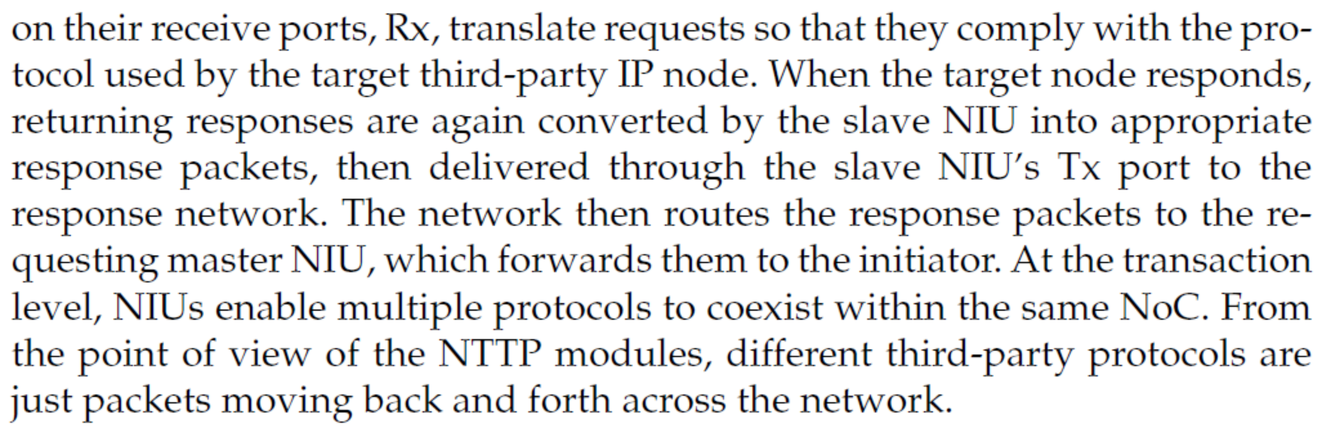
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Arteris Interconnect IP Solution Selected by Samsung for Mobile SoC Deployment by **Kurt Shuler**, on November 02, 2010 Network-on-Chip (NoC) interconnect technology leader enables higher performance and more cost effective designs for mobile phone systems-on-chip (SoCs) SUNNYVALE, California — November 2, 2010 — Arteris, Inc., a leading supplier of on-chip interconnect IP solutions, today announced that Samsung Electronics Co., Ltd., has selected Arteris' interconnect solutions for multiple chips within Samsung's mobile SOC products. Samsung chose Arteris interconnect IP to support the high speed inter-chip communication requirements in next generation mobile SOC products. *The Arteris interconnect IP offers us a convenient solution to handle the high speed communication needed between our SoC and external modem IC. Our customers will benefit from the lower BOM cost and power consumption as a result of this IP. We look forward to Arteris' interconnect IP helping us shorten development schedules and lower risks associated with compatibility.* *Thomas Kim, Vice President, SoC Platform Development, System LSI,* ***Samsung Electronics*** https://www.arteris.com/press-releases/pr_2010_nov_02?hsLang=en-us A large SoC, such as the Exynos SoC may include multiple classes of Arteris NoC interconnect:

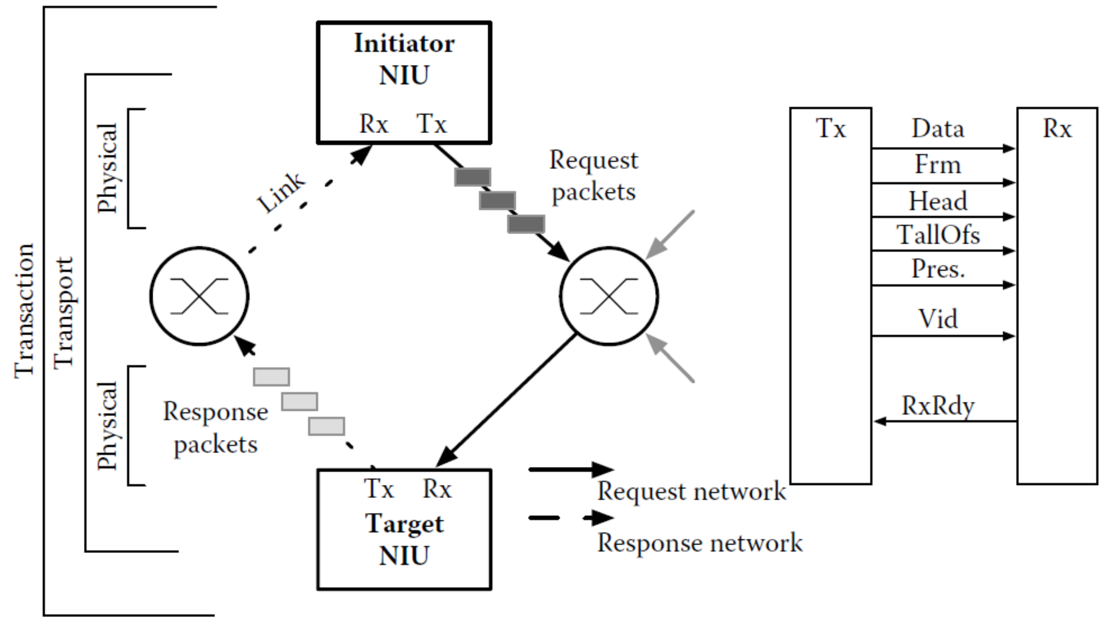
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Logical Interconnect Topology Development FLEXNOC & NCORE INTERCONNECT IPS DEFINE ARCHITECTURES Main NoC Ncore Cache Coherent NoC Service NoC Memory NoC Video NoC Cache Coherent Interconnect Main Interconnect Memory NoC Video NoC CSR NoC V voltage P power A async Q QoS F firewall CPU big CPU little GPU DSP ZRAM Coherent Subsystem DIRU 0 DIRU 1 CMIU D0 CMIU D1 SRAM DRAM Sched 0 Sched 1 Sched 2 Sched 3 • ArChip16 Example: Large SoCs have multiple classes of interconnect – Non-coherent, Coherent, Control/Status, Observability, etc. • Ncore & FlexNoC interconnects are managed separately from IP blocks, increasing design flexibility ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 9 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9. The Arteris NoC is an interconnect connects the plurality of processing modules in the Exynos SoC through interface units, wherein a first processing module communicates to a second processing module using transactions.

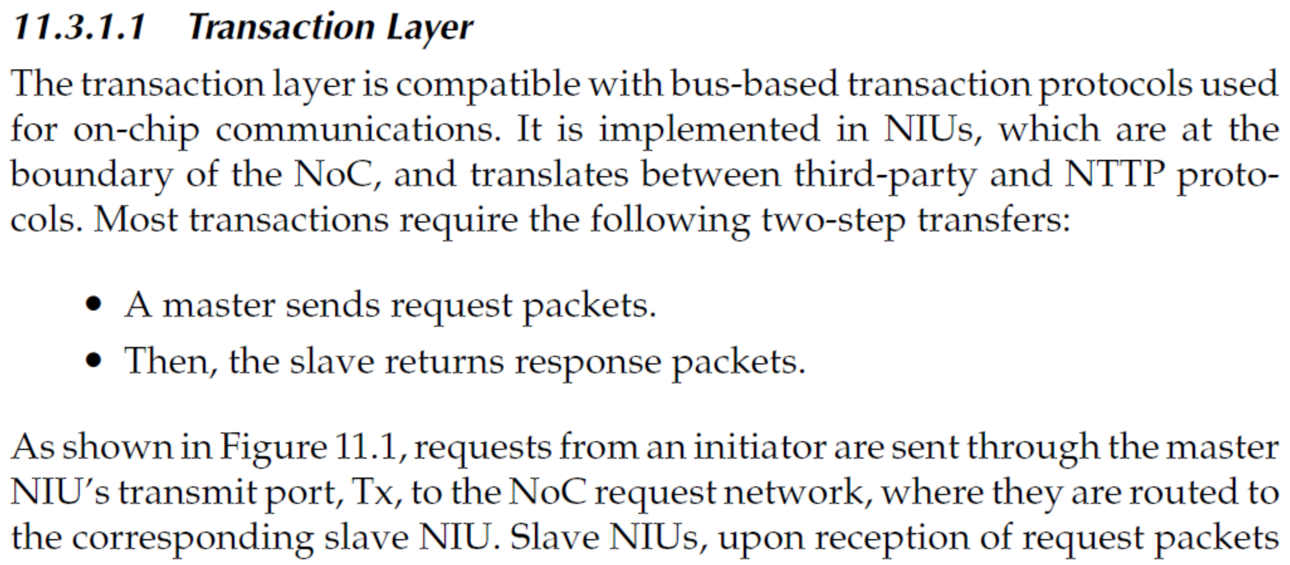
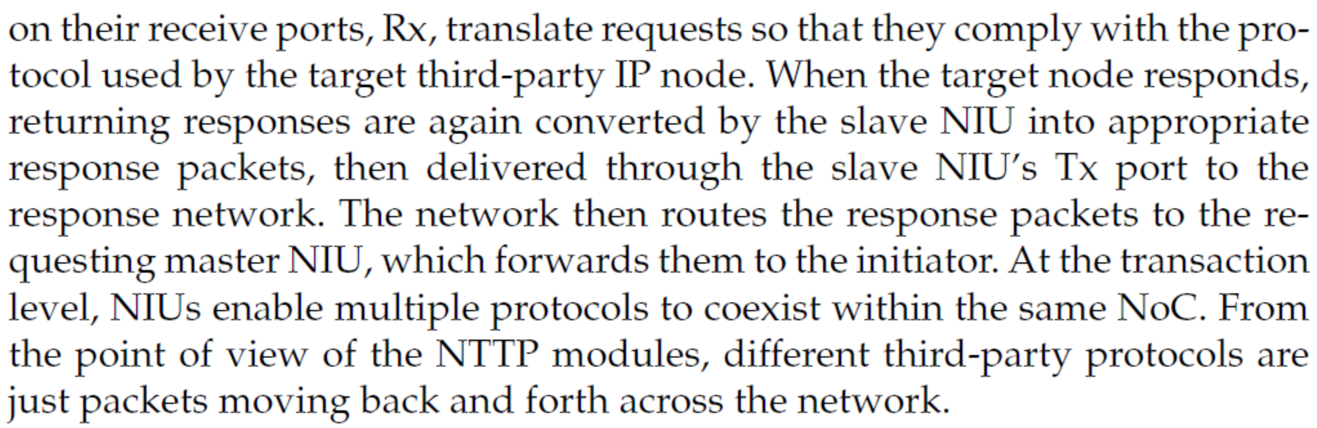
4855-4585-1715.1

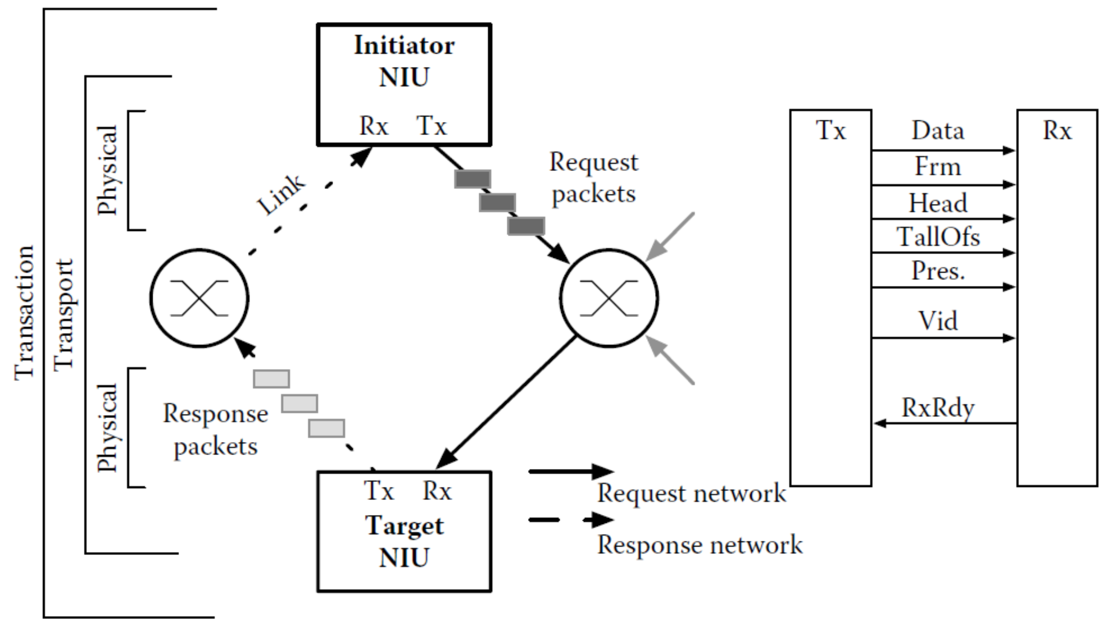
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

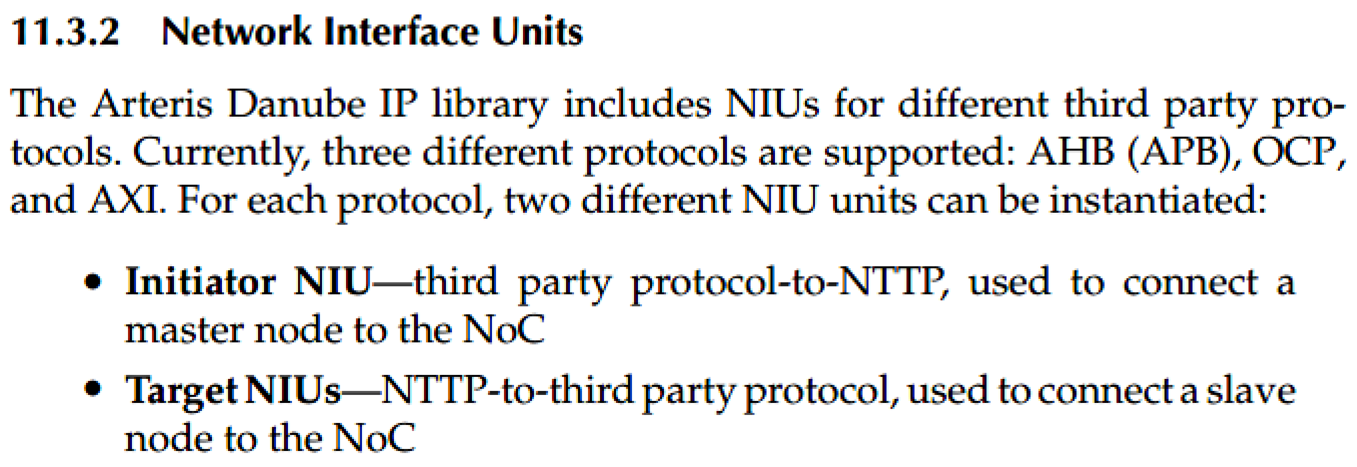
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.
configuring the first processing module having a first memory as a master the provides requests;	The Arteris NoC utilized by the Exynos SoC configures the first processing module having a first memory as a master the provides requests, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

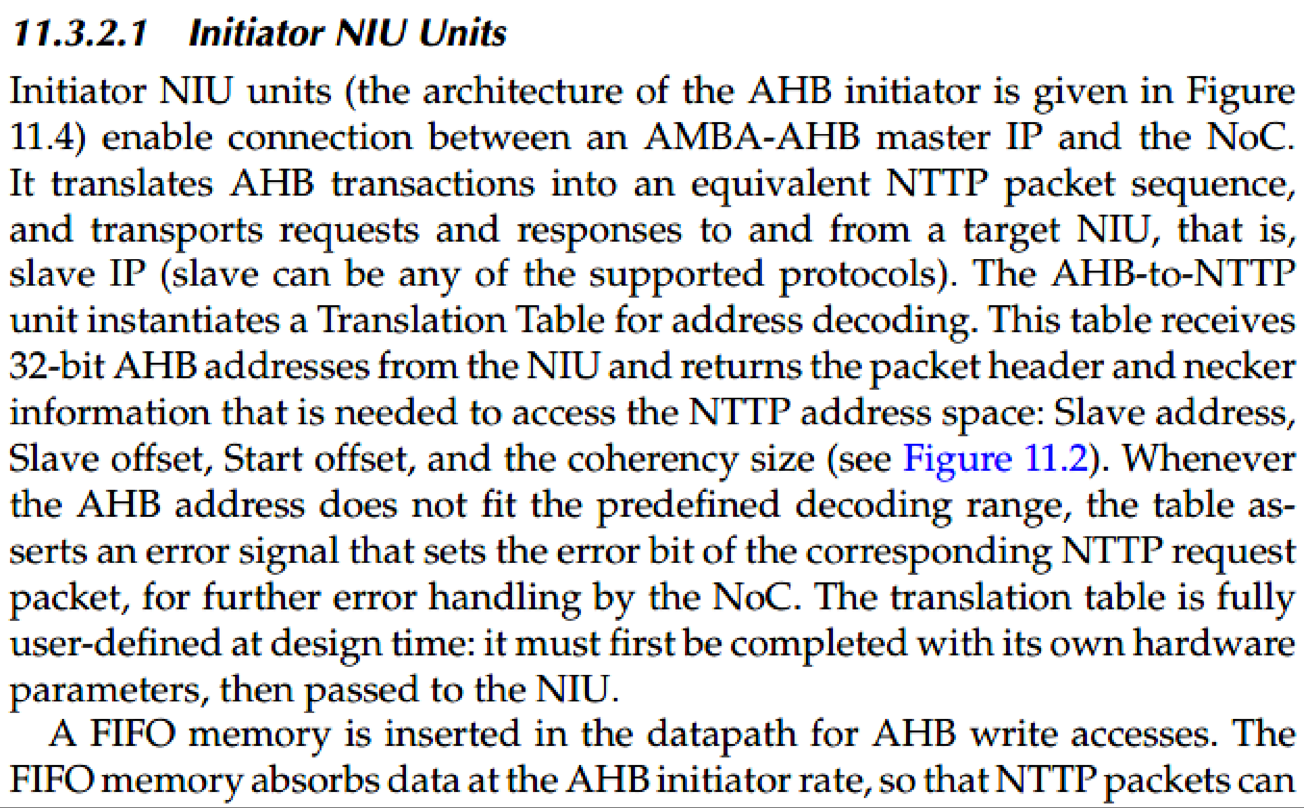
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets  on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

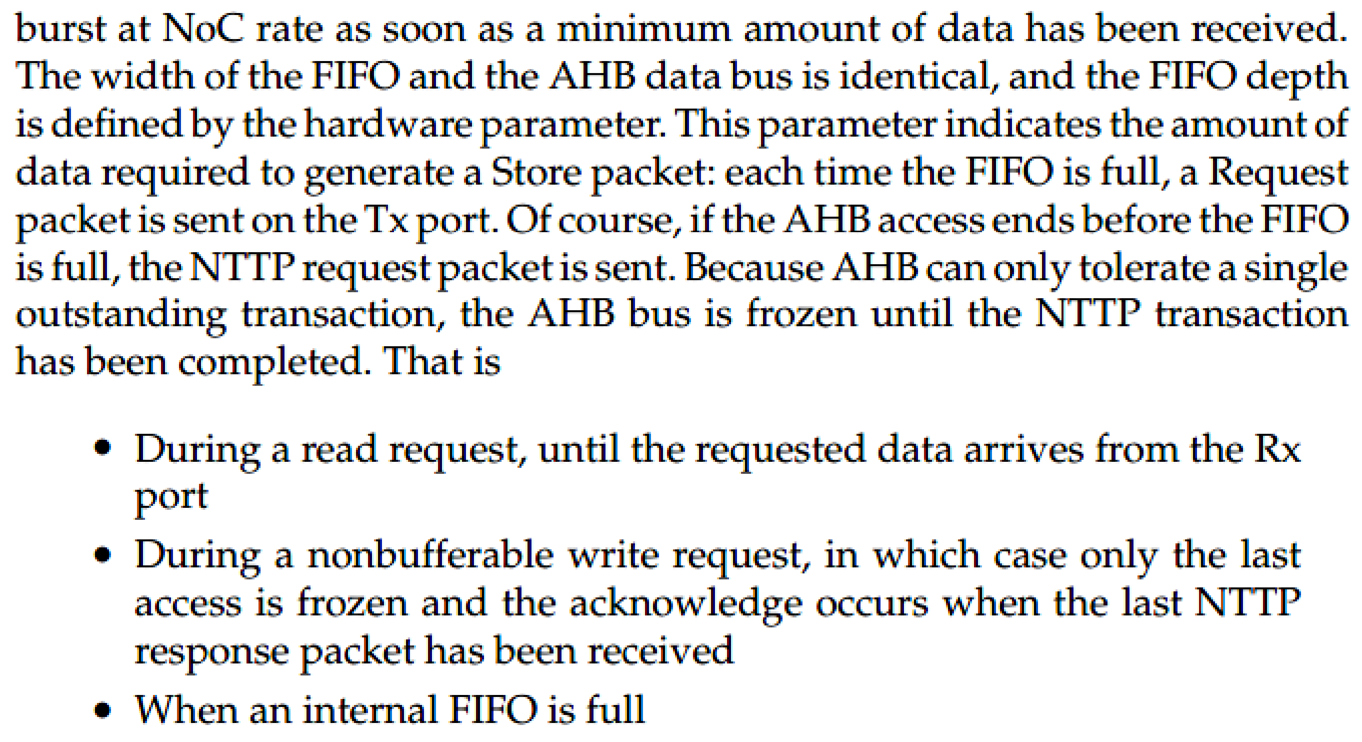
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

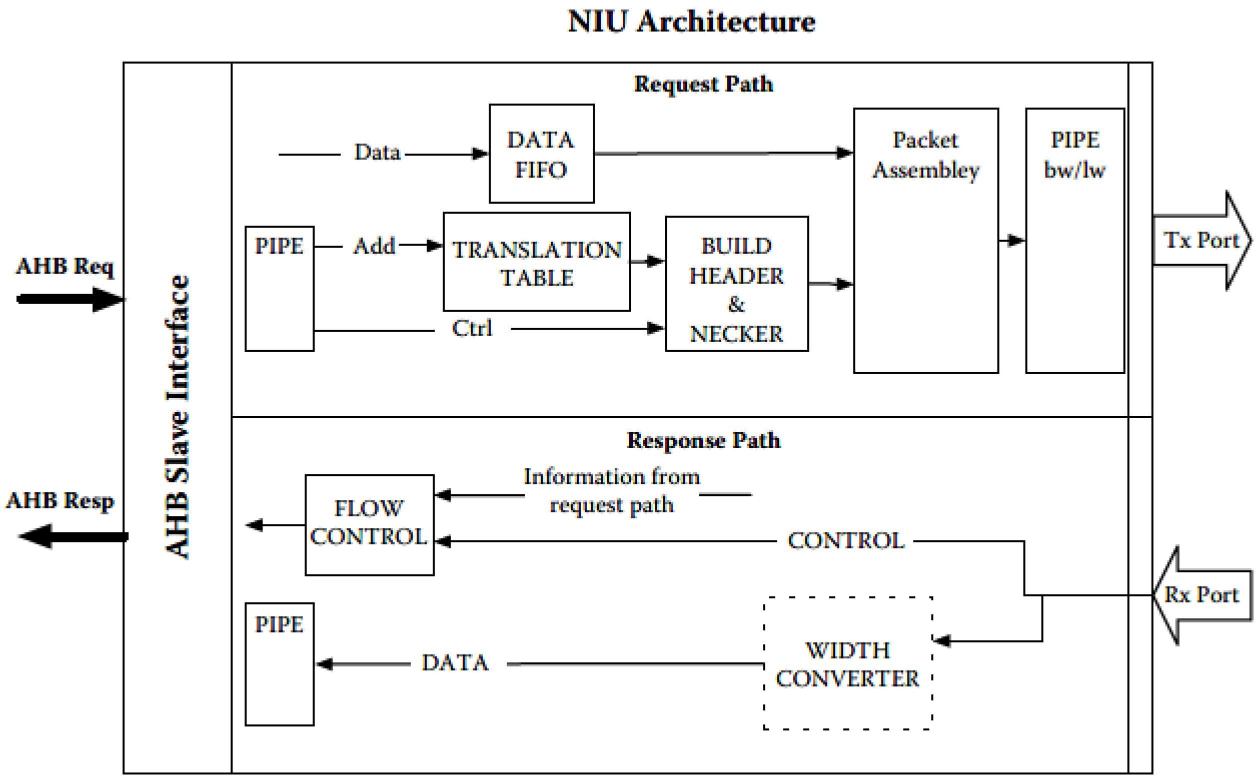
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

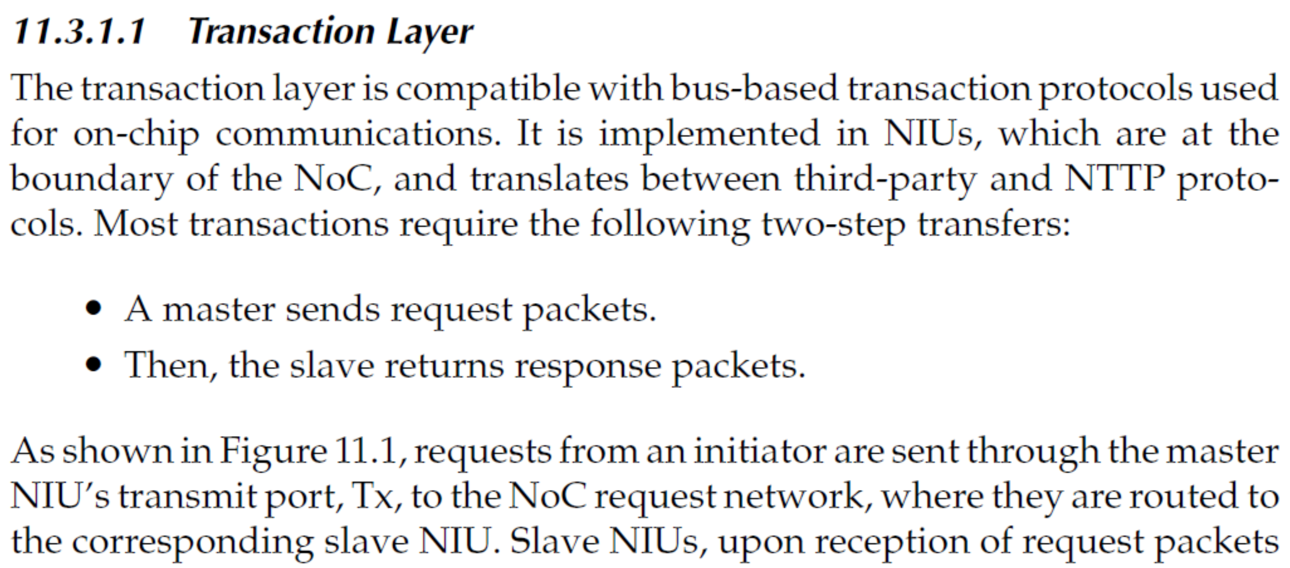
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC [and] translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and has a "FIFO memory […] inserted in the datapath for AHB write access":

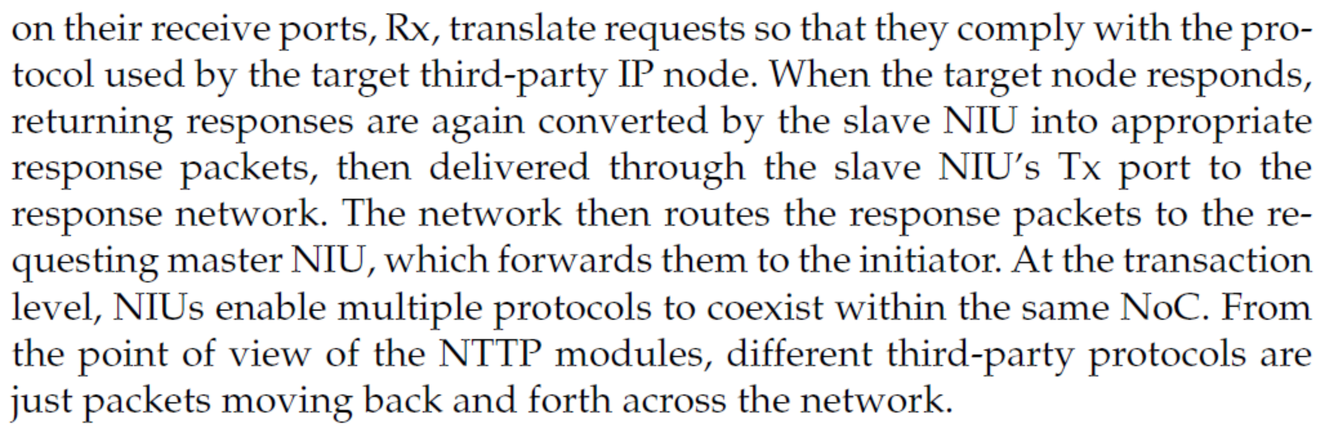
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

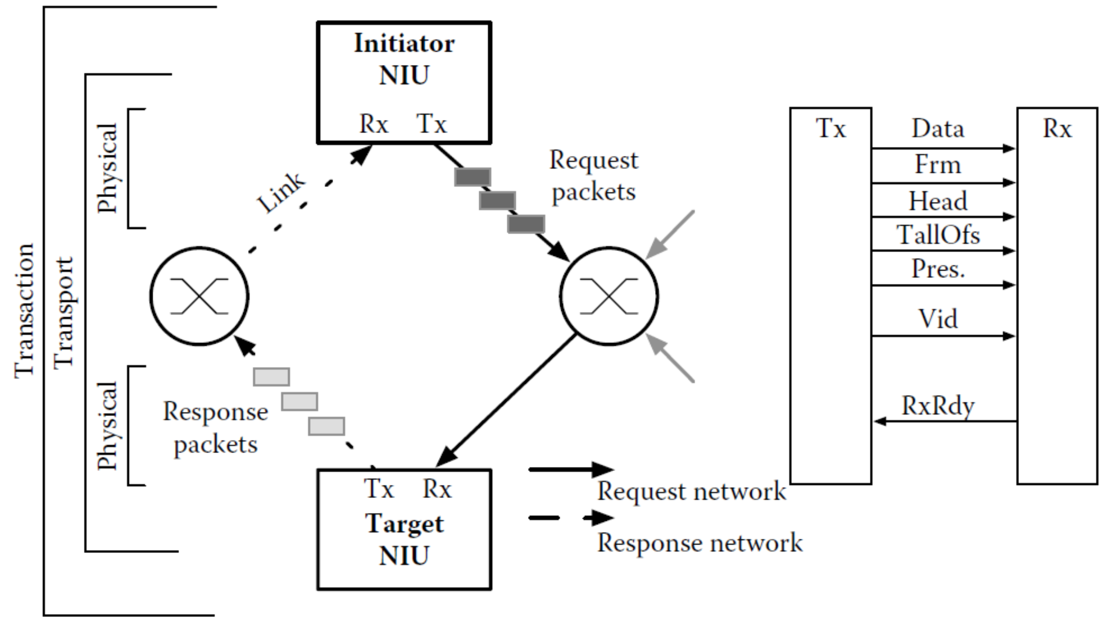
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

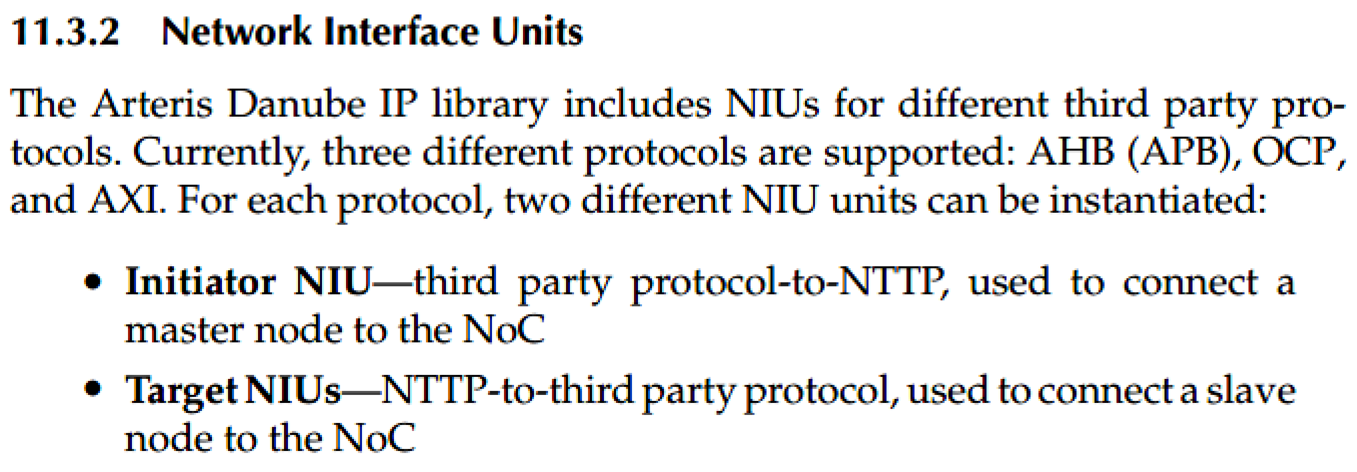
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl AHB Req AHB Slave Interface Tx Port Response Path AHB Resp FLOW CONTROL Information from request path CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
configuring the second processing module having a second memory as	The Arteris NoC utilized by the Exynos SoC configures the second processing module having a second memory as a slave the provides responses to the requests, either literally or under the doctrine of equivalents.

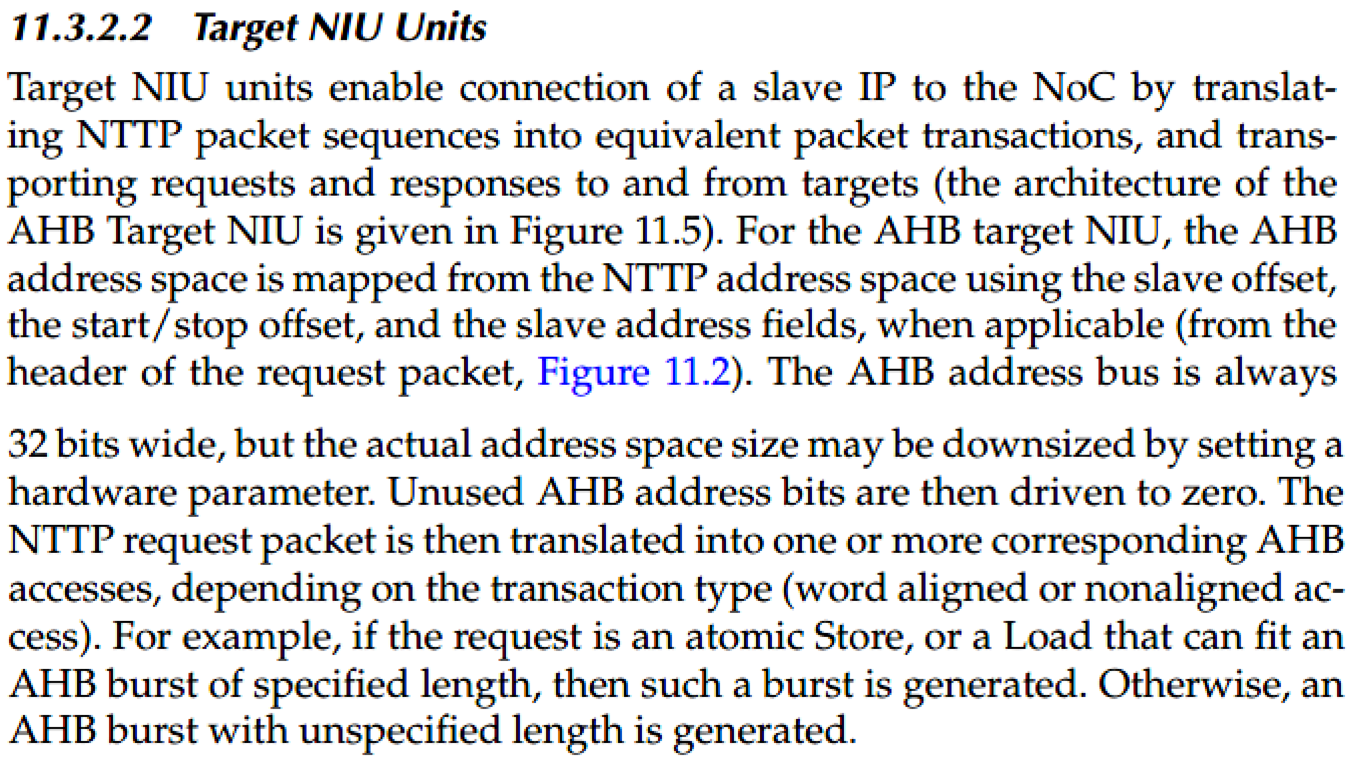
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
a slave the provides responses to the requests;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

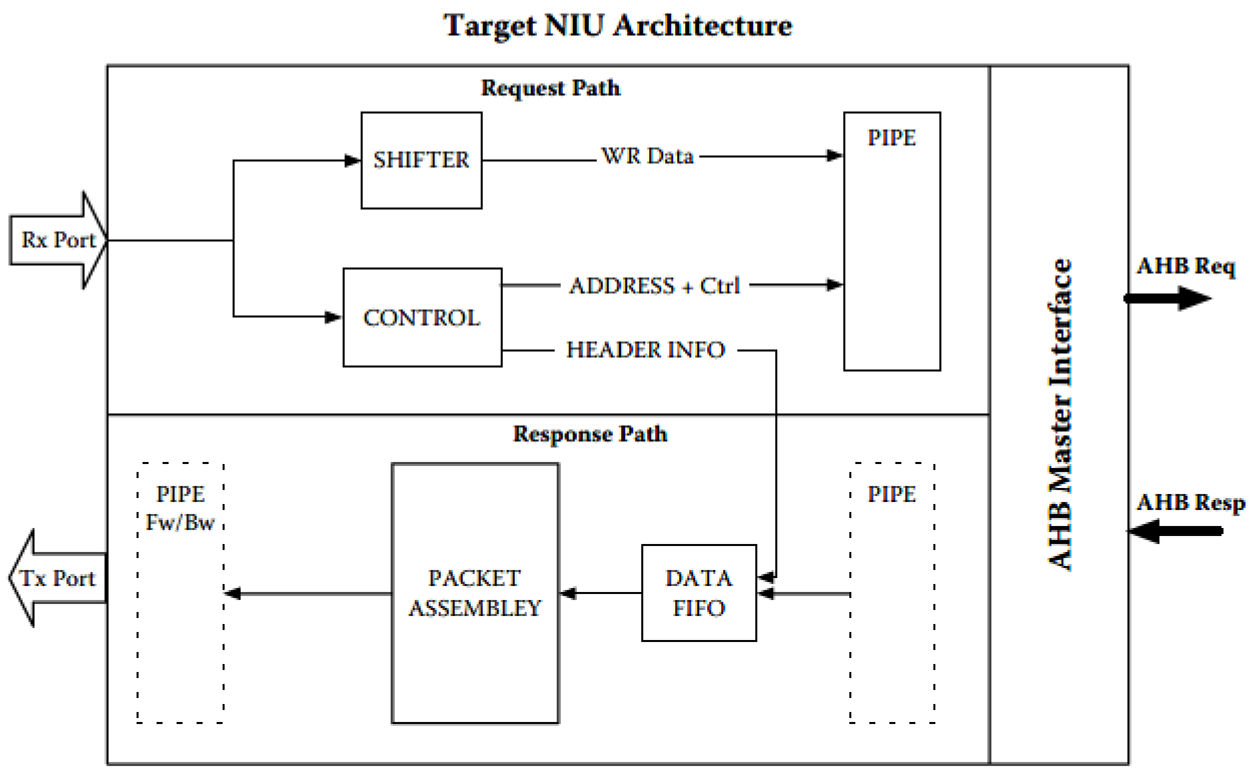
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

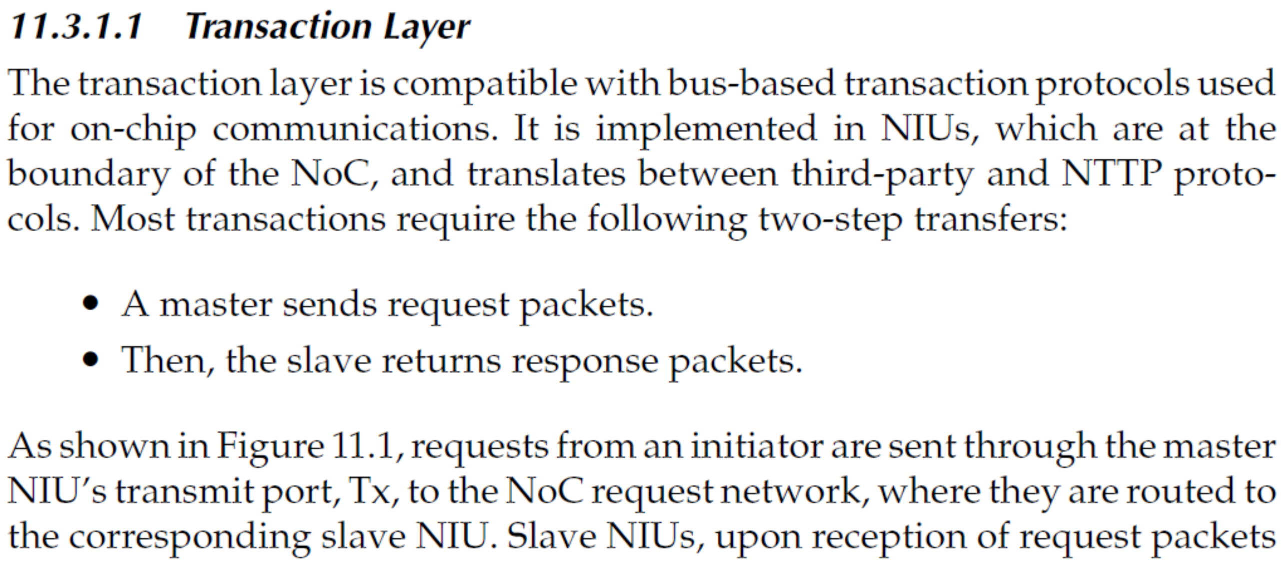
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

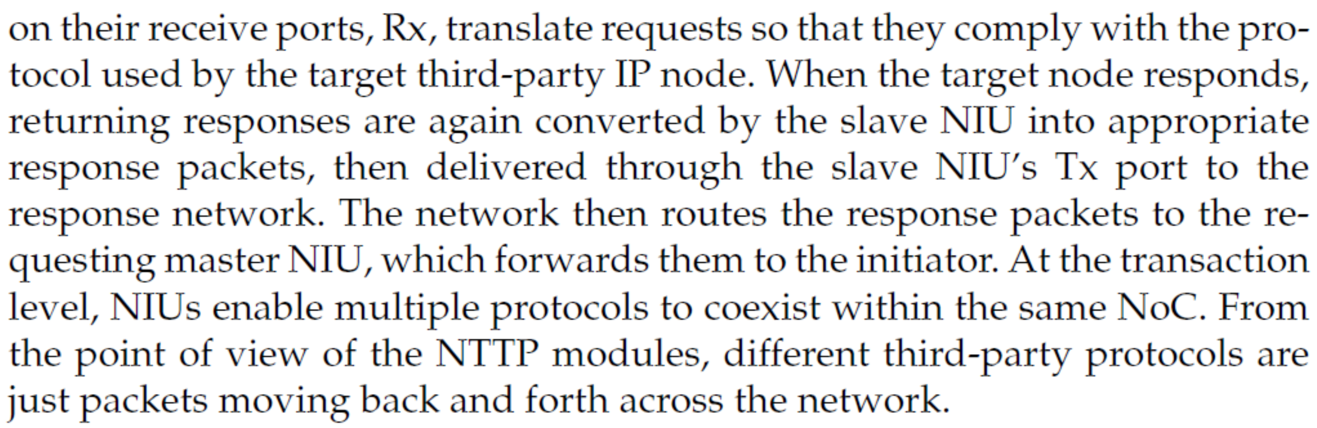
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and have a FIFO memory in the datapath:

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

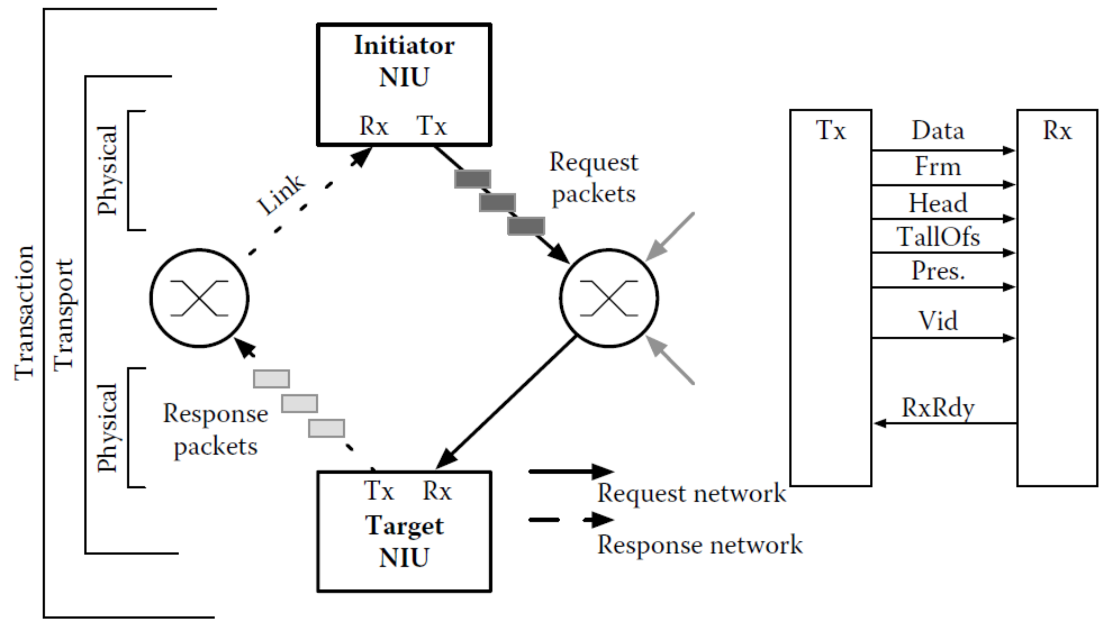
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Target NIU Architecture Request Path SHIFTER WR Data PIPE Rx Port CONTROL ADDRESS + Ctrl HEADER INFO AHB Master Interface AHB Req Response Path PIPE Fw/Bw PACKET ASSEMBLEY DATA FIFO PIPE Tx Port AHB Resp **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.
connecting the master to a master	The Arteris NoC utilized by the Exynos SoC connects the master to a master interface unit of the interface units, either literally or under the doctrine of equivalents.

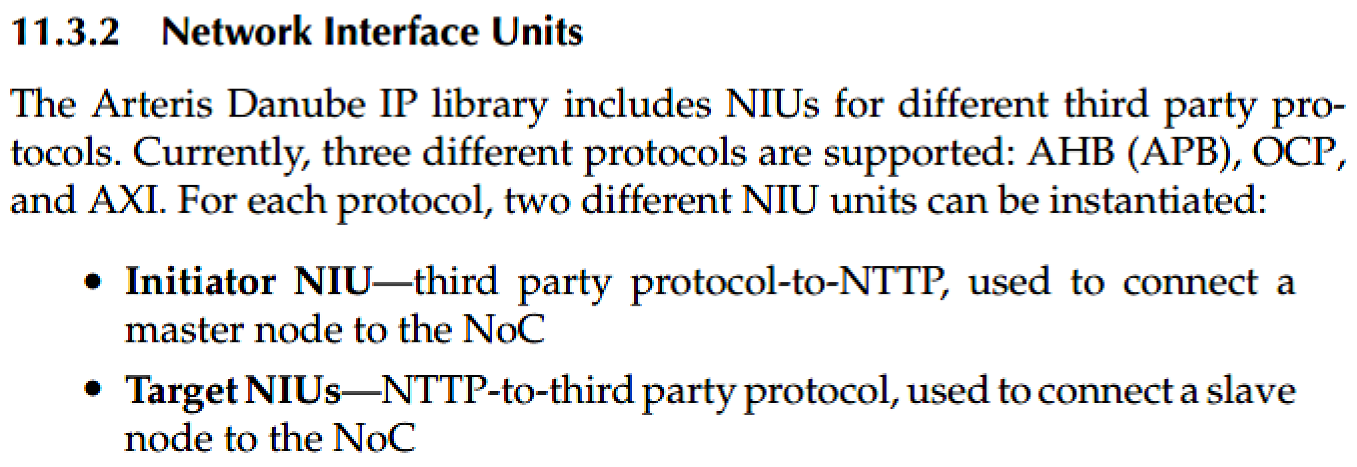
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
interface unit of the interface units;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

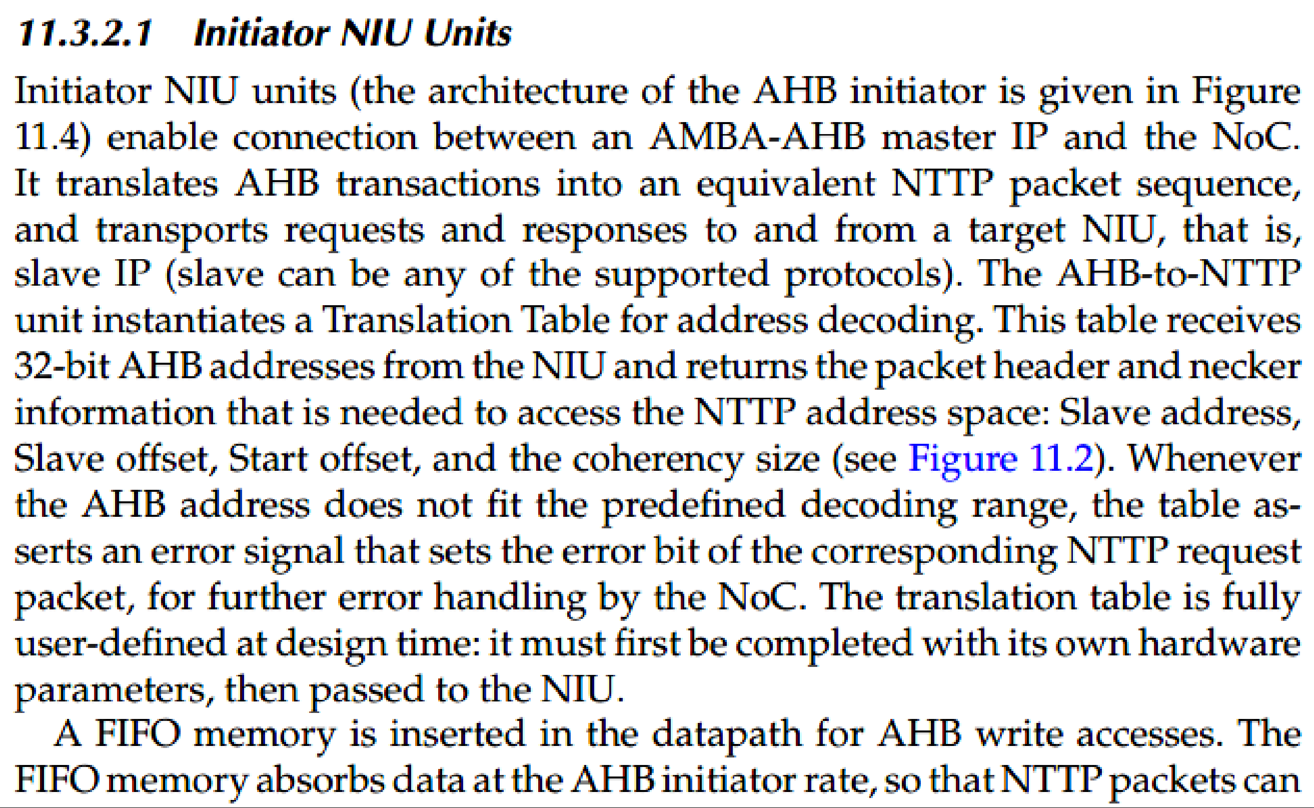
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

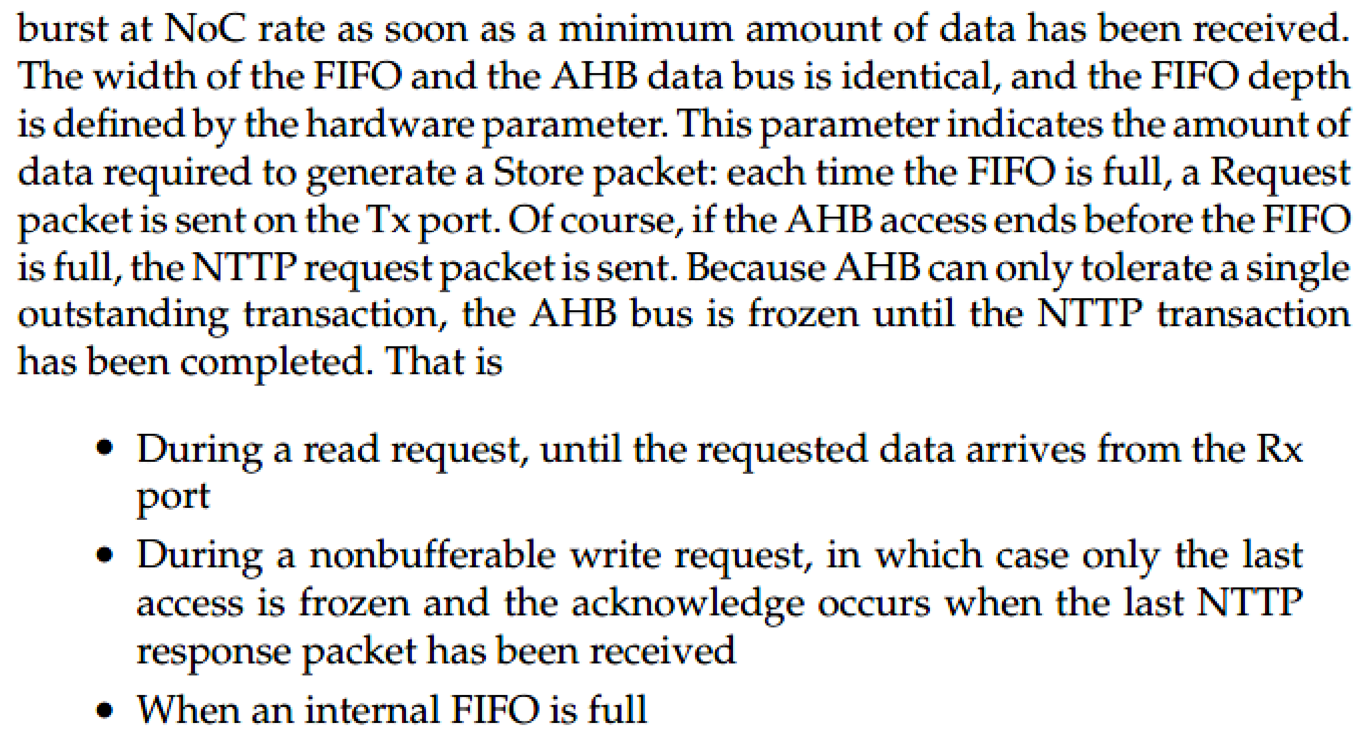
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Transaction Transport Physical Physical Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC":

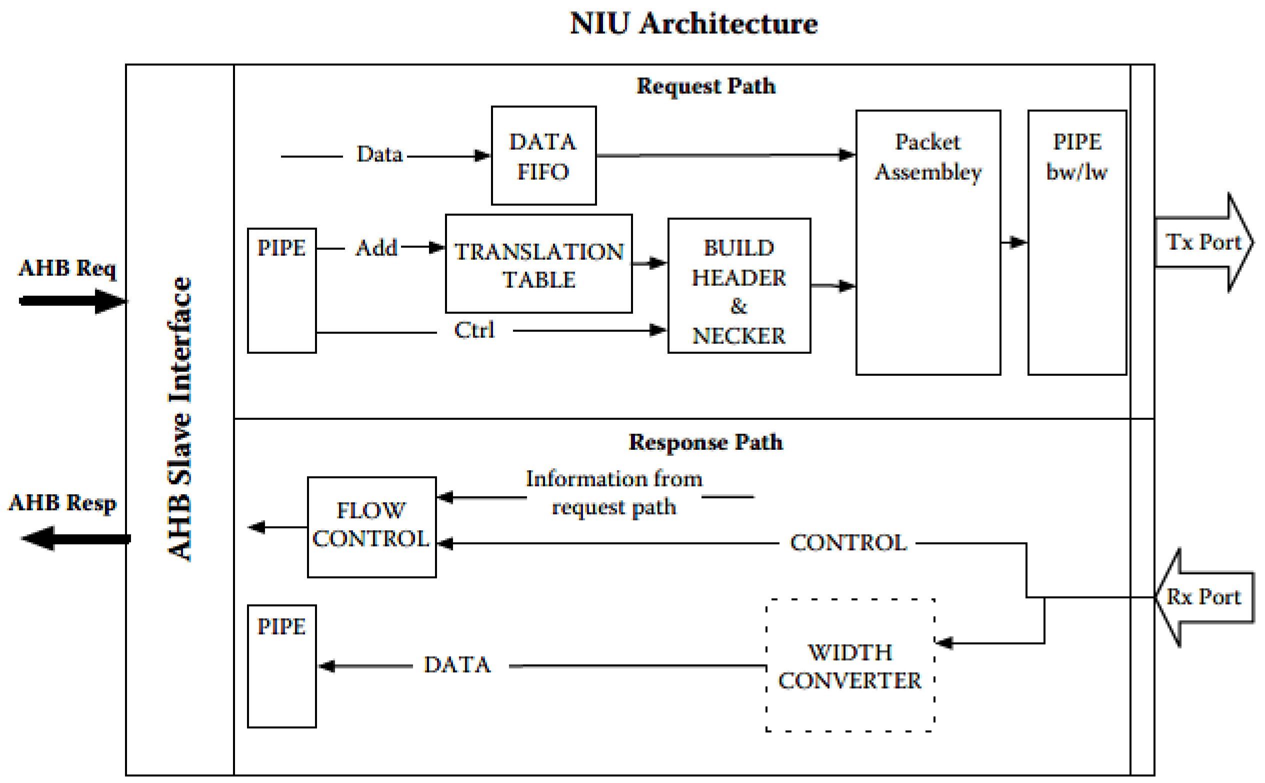
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

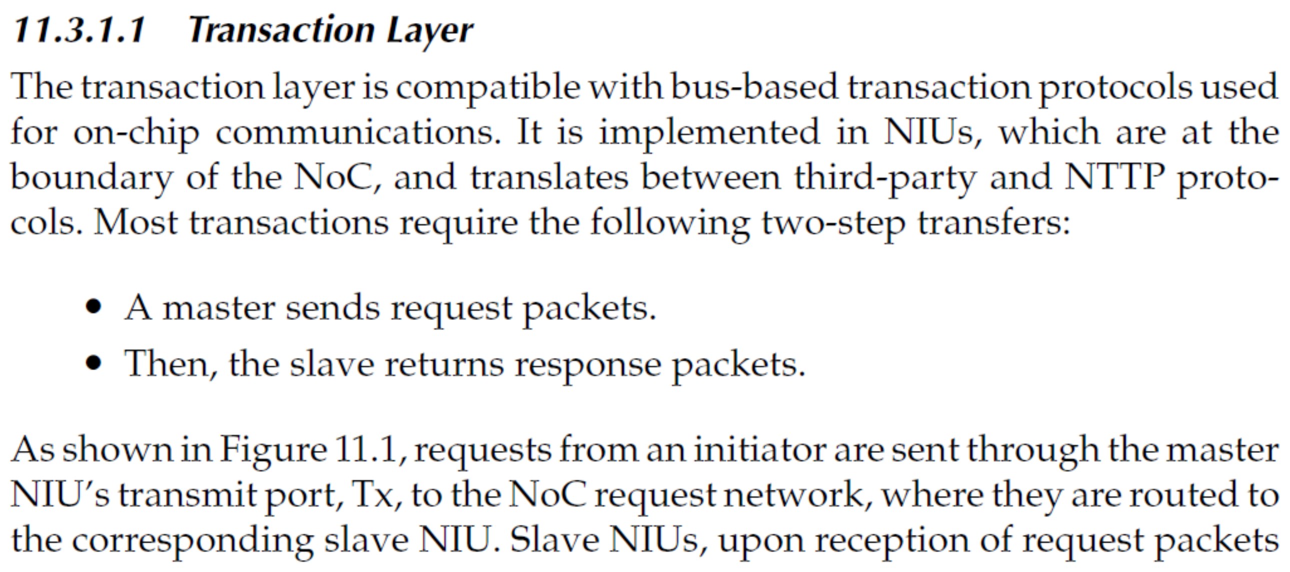
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

4855-4585-1715.1

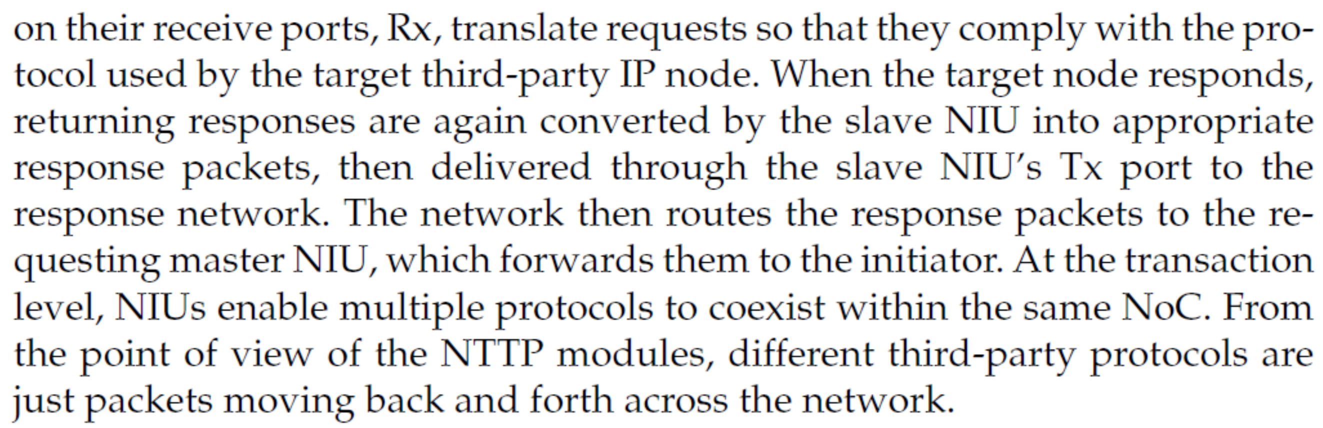
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

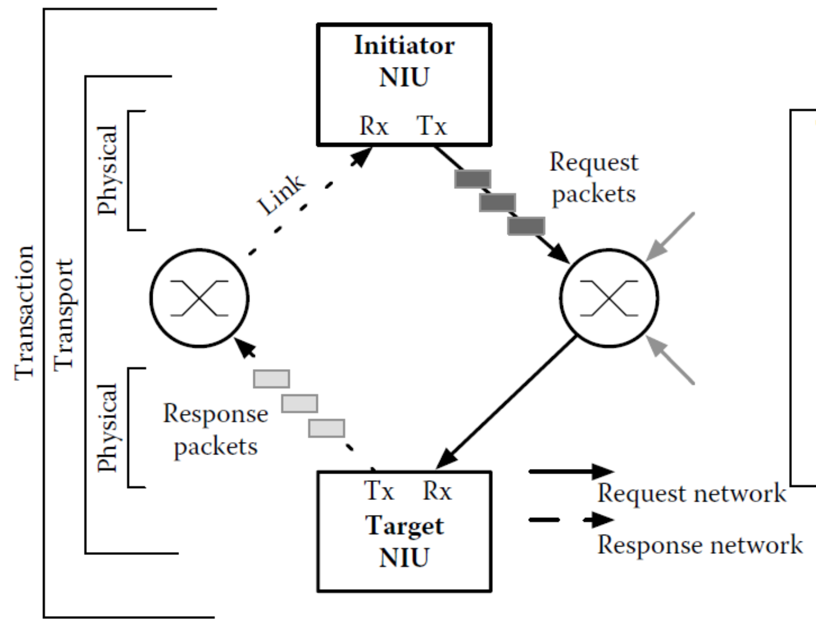
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface AHB Resp Response Path Information from request path FLOW CONTROL CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
connecting the master interface unit to the	The Arteris NoC utilized by the Exynos SoC connects the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect, either literally or under the doctrine of equivalents.

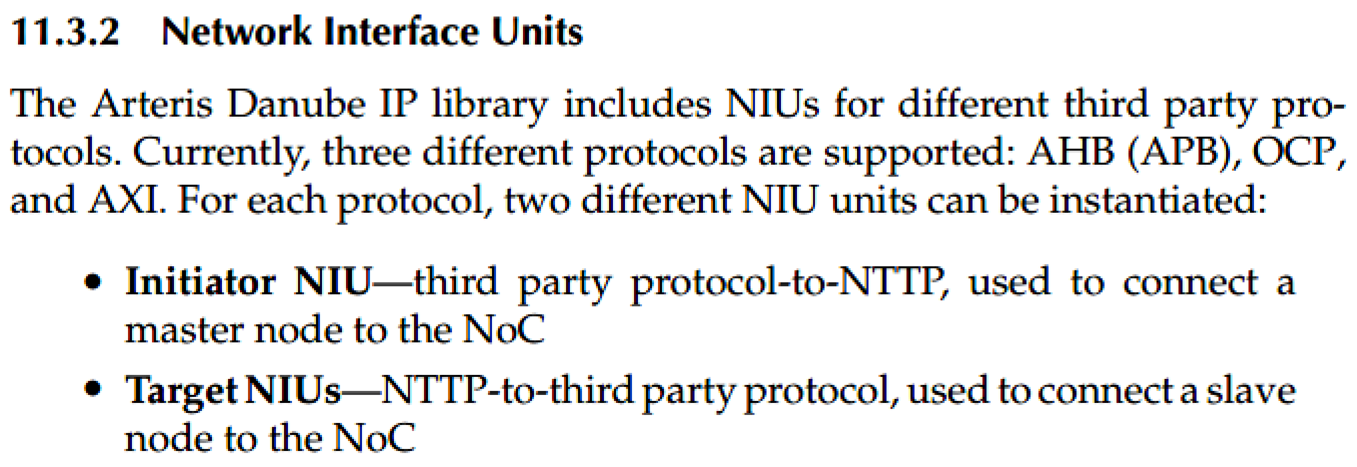
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
interconnect so that the master interface unit is between the master and the interconnect;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

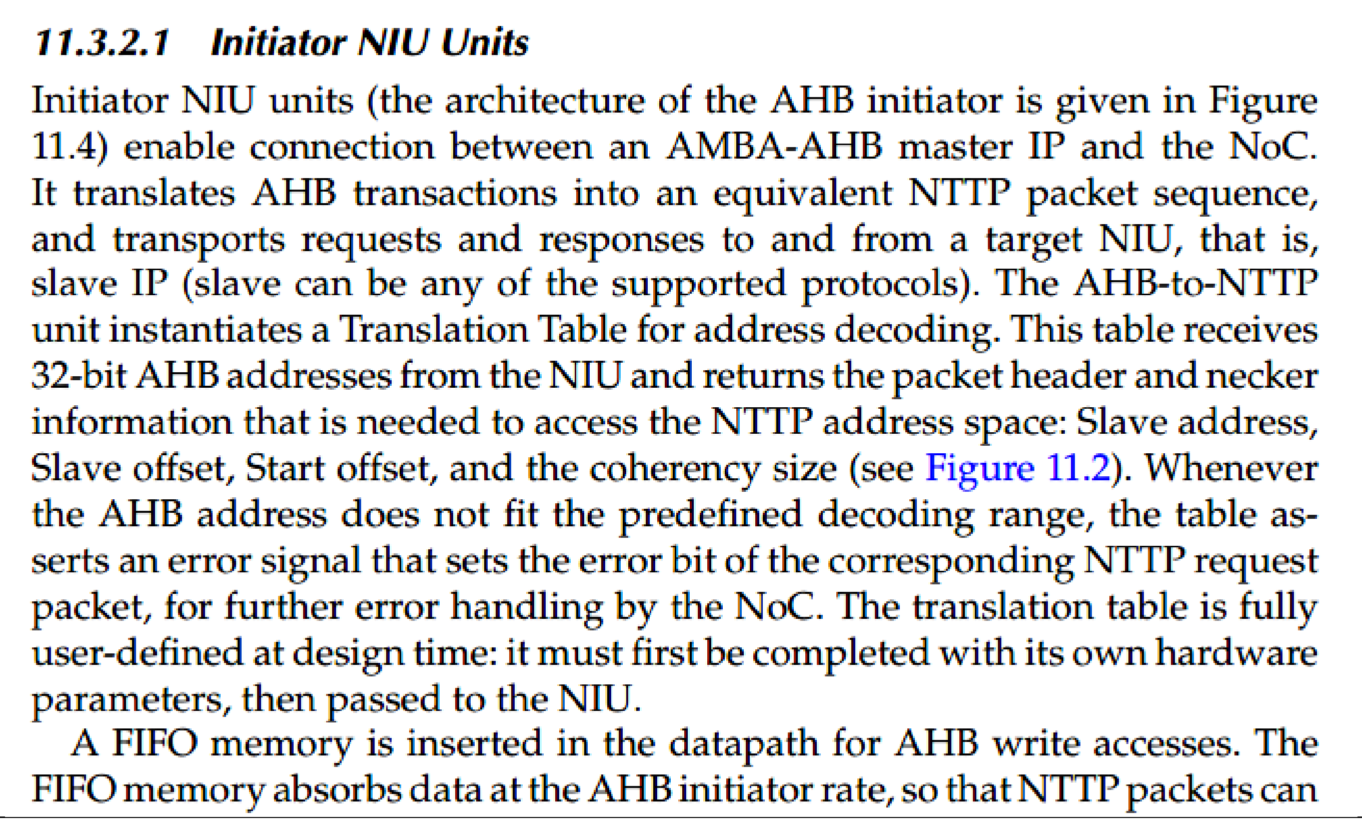
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

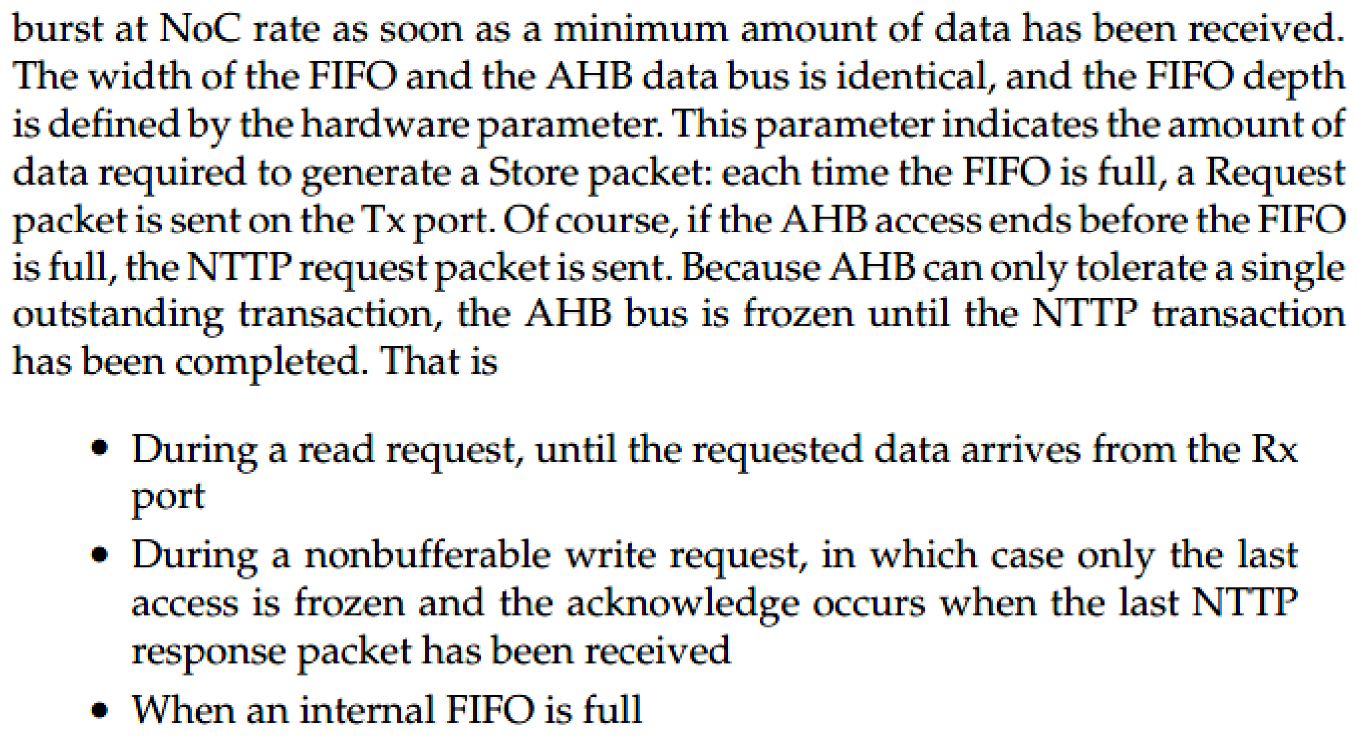
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Initiator NIU Rx Tx Link Request packets Transaction Transport Physical Physical Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

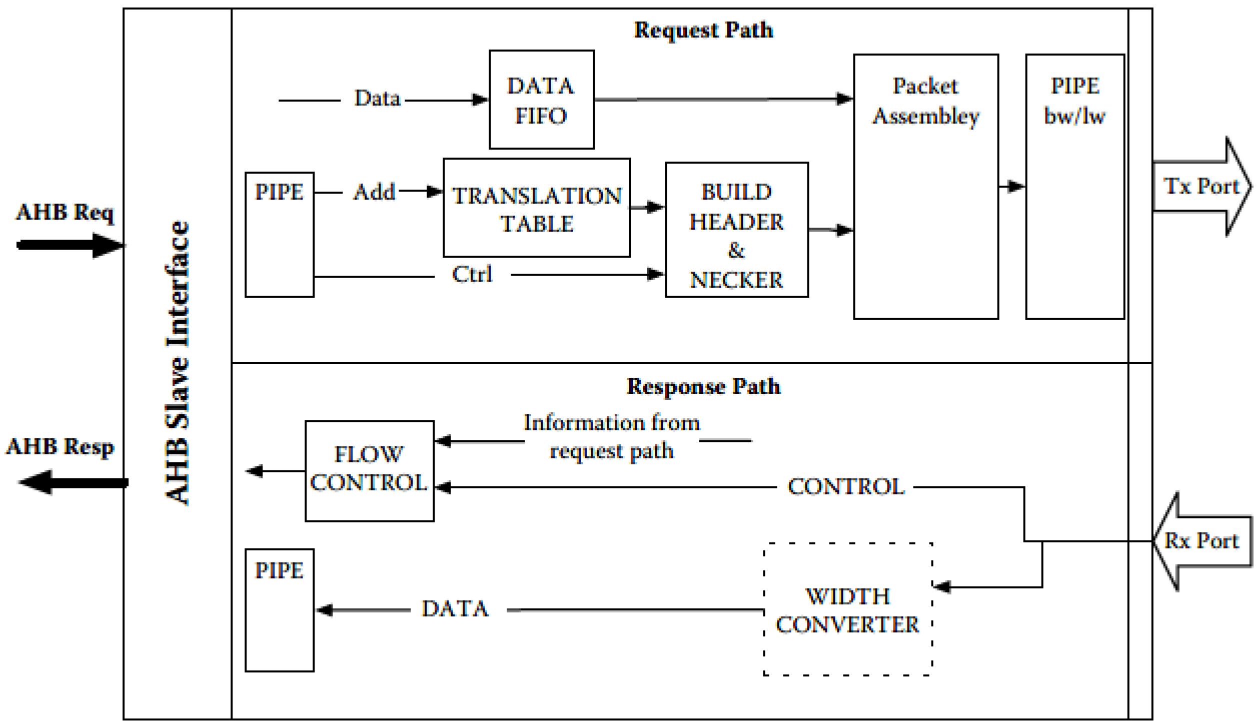
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

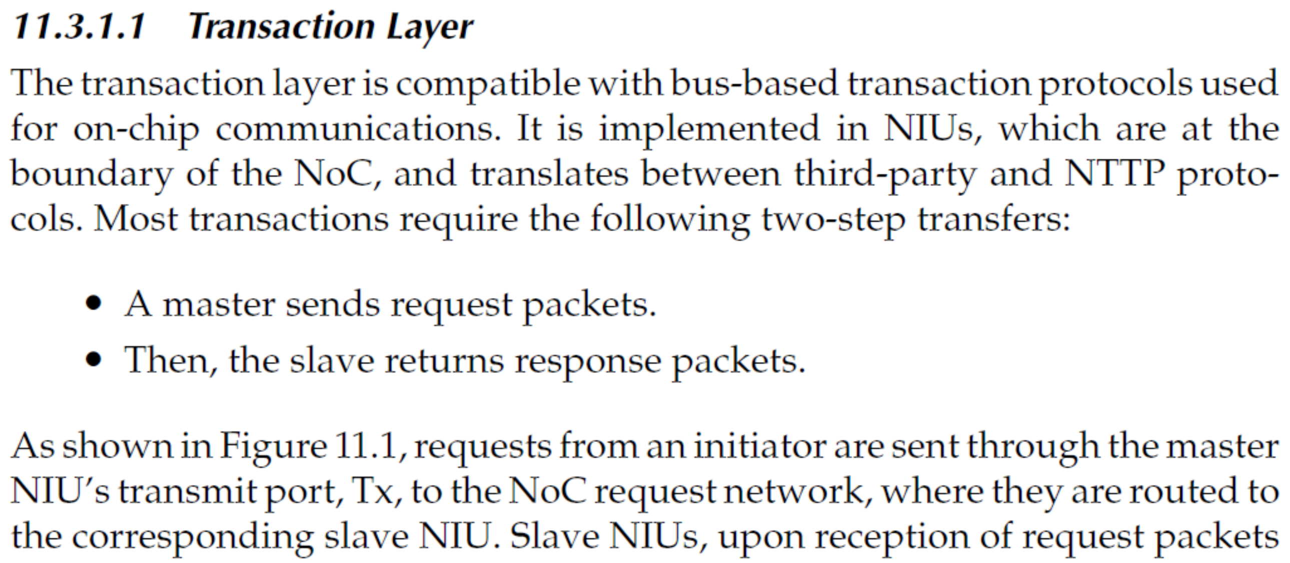
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

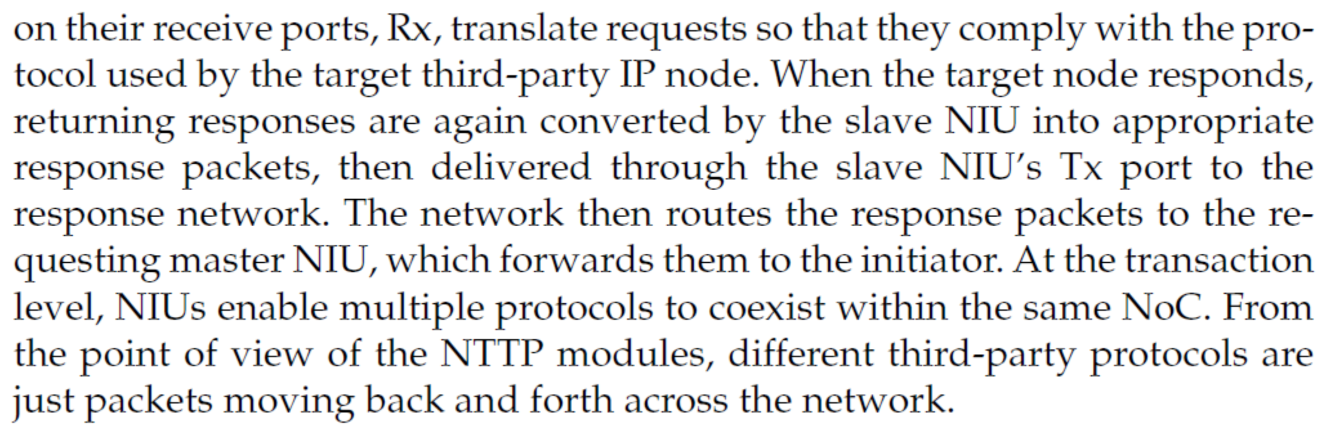
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2.1 Initiator NIU Units** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

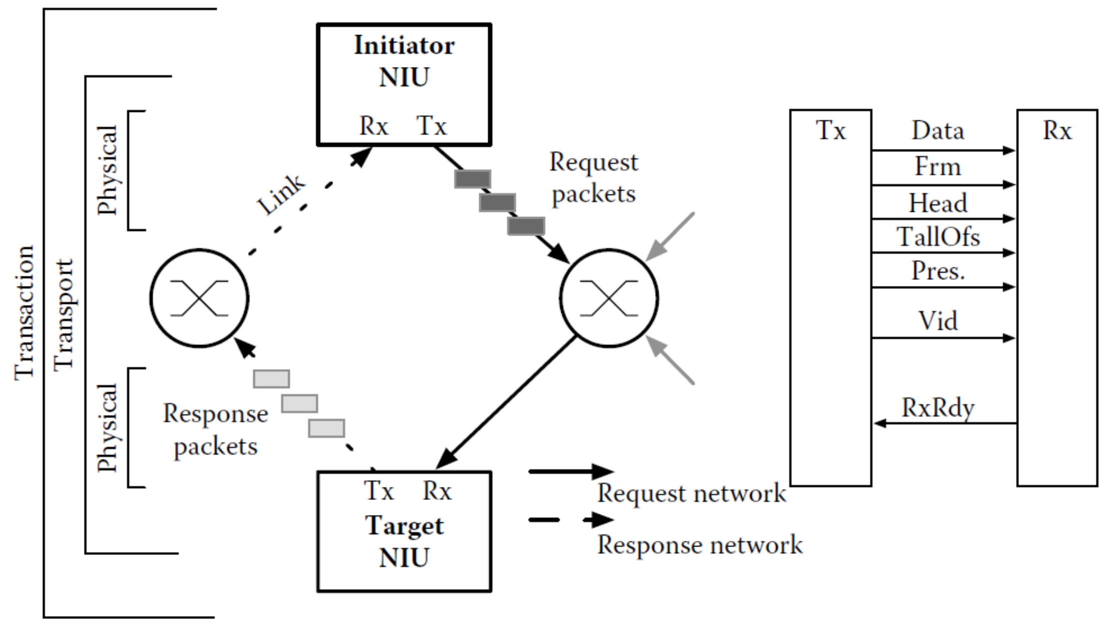
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	NIU Architecture  Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path Information from request path AHB Resp FLOW CONTROL CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
connecting the slave to a slave	The Arteris NoC utilized by the Exynos SoC connecting the slave to a slave interface unit of the interface units, either literally or under the doctrine of equivalents.

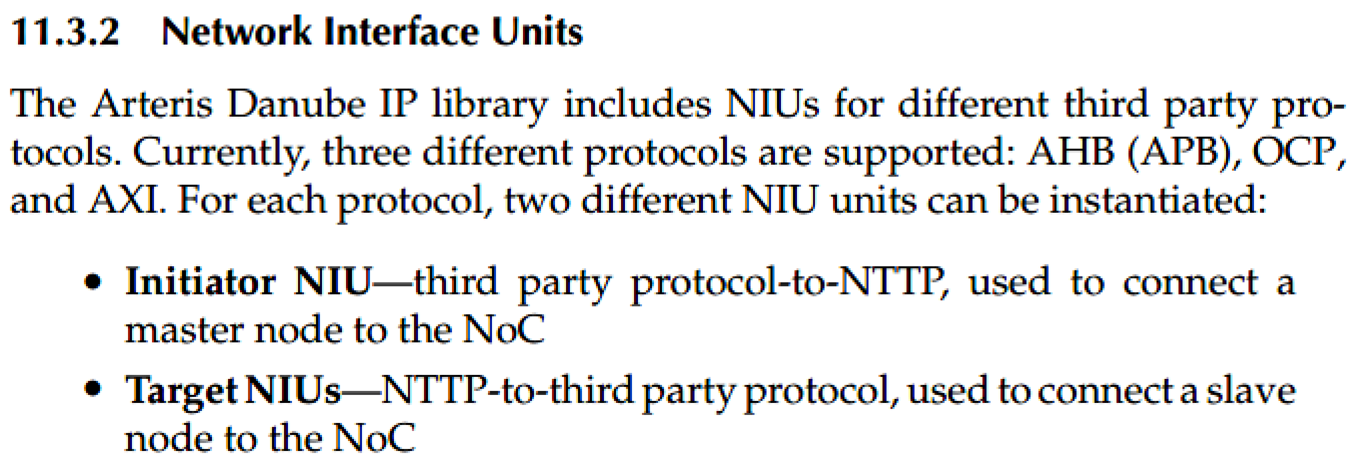
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
interface unit of the interface units;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

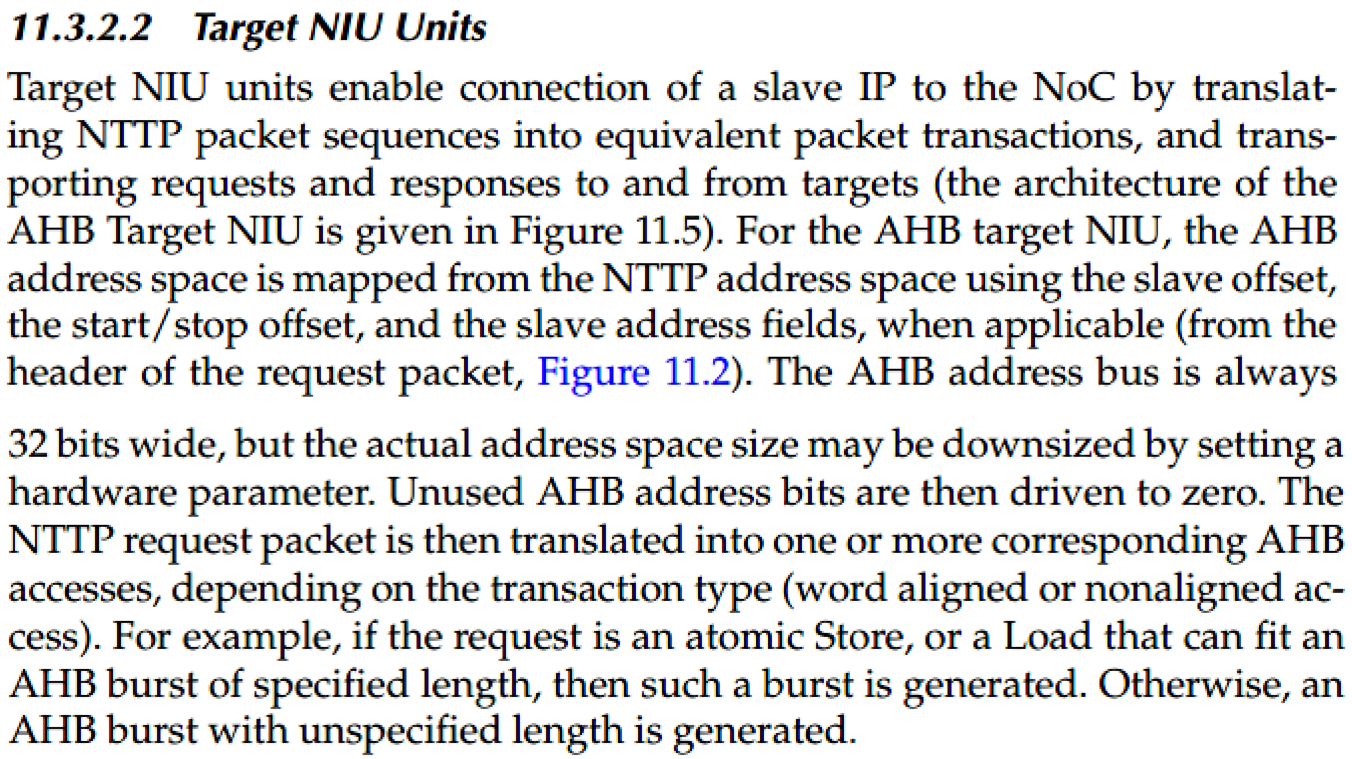
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

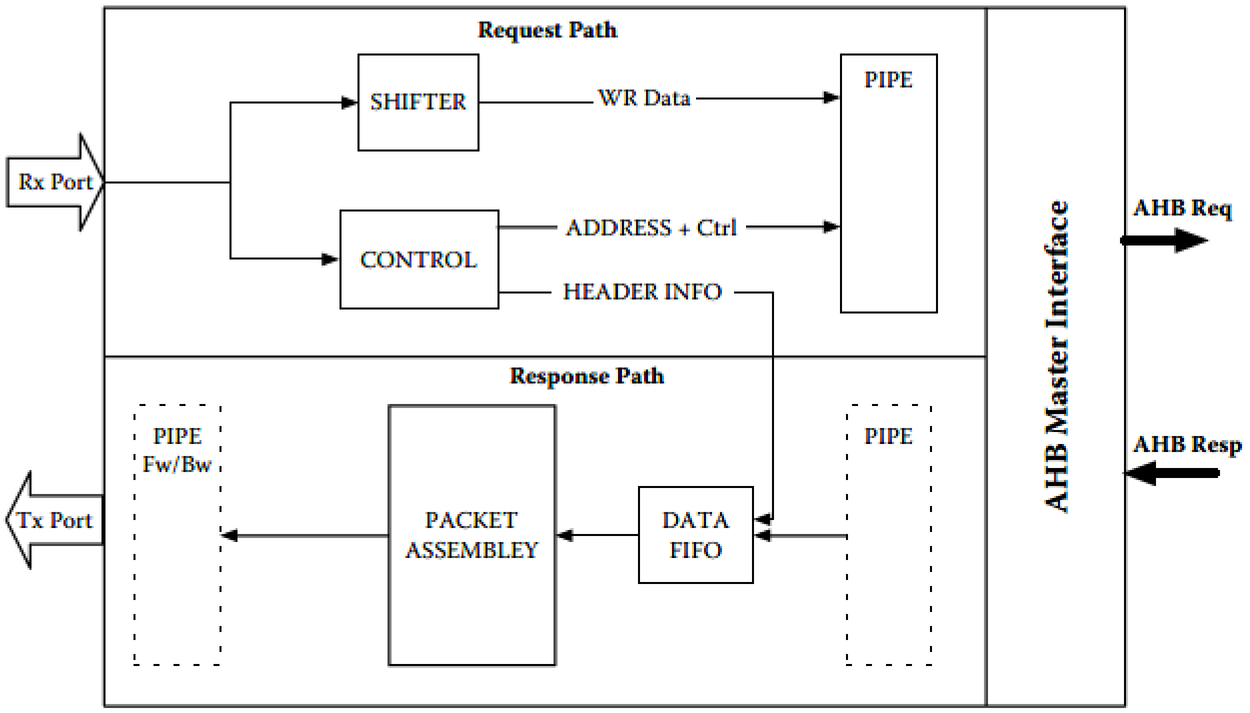
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

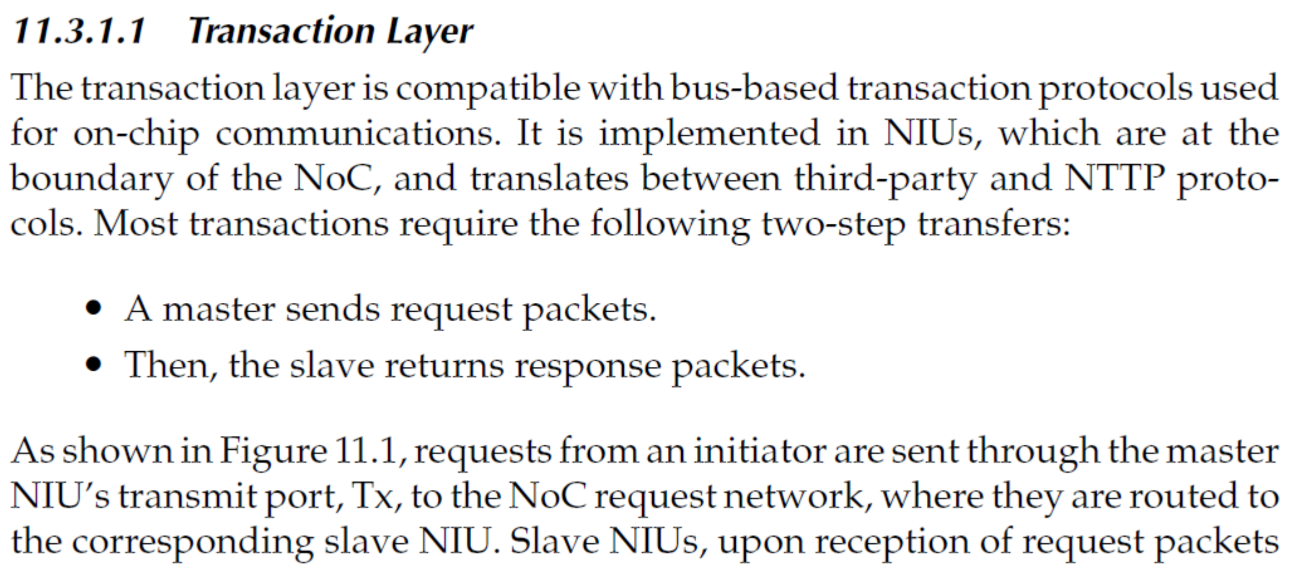
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Initiator NIU Rx Tx Link Request packets Transaction Transport Physical Physical Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

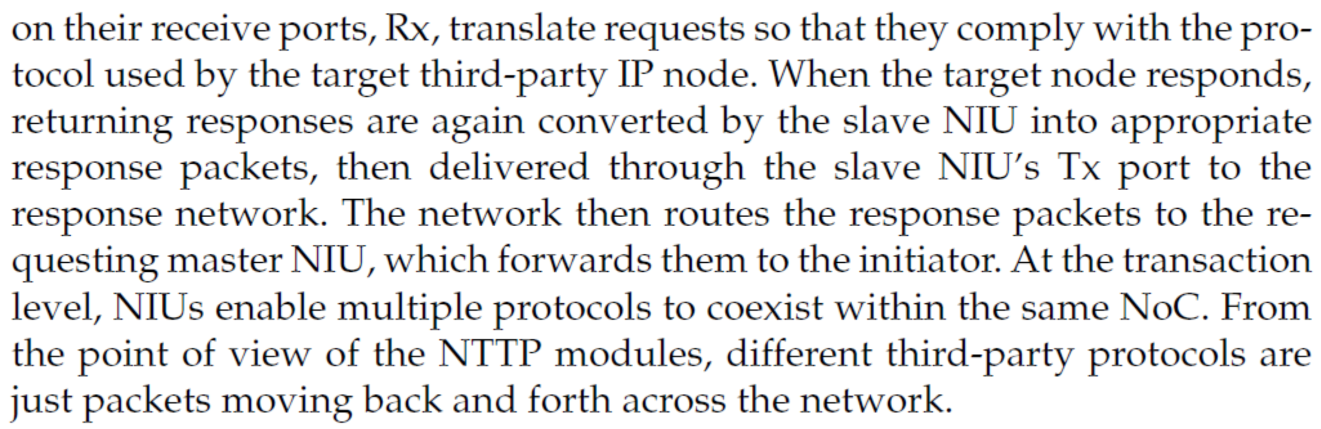
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets":

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

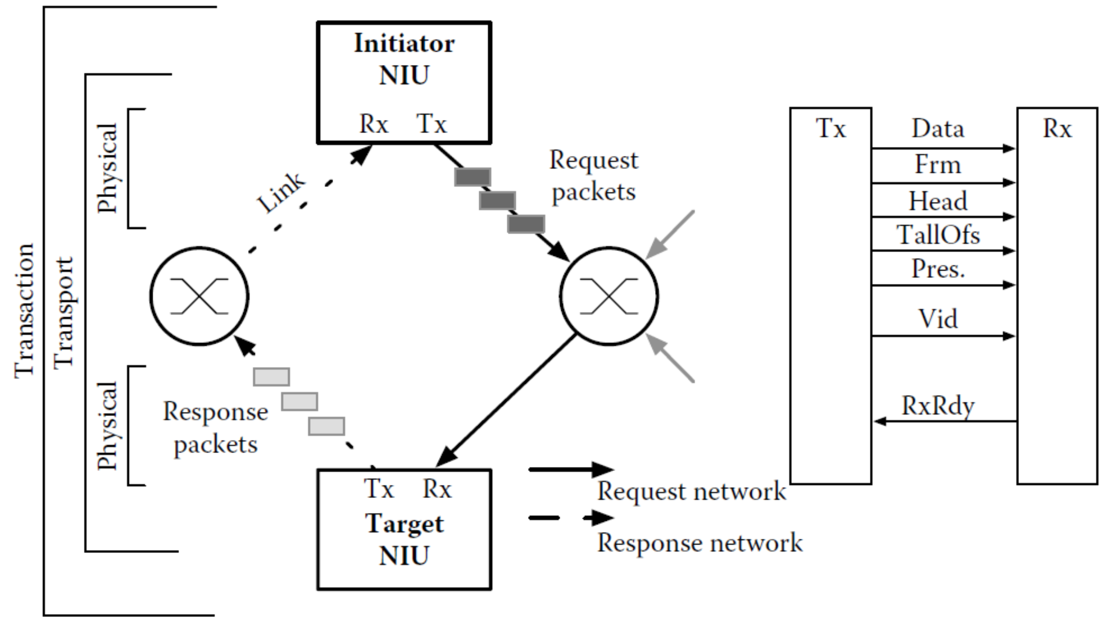
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	Target NIU Architecture  Request Path SHIFTER WR Data PIPE Rx Port CONTROL ADDRESS + Ctrl HEADER INFO AHB Master Interface AHB Req AHB Resp Response Path PIPE Fw/Bw PIPE Tx Port PACKET ASSEMBLEY DATA FIFO **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

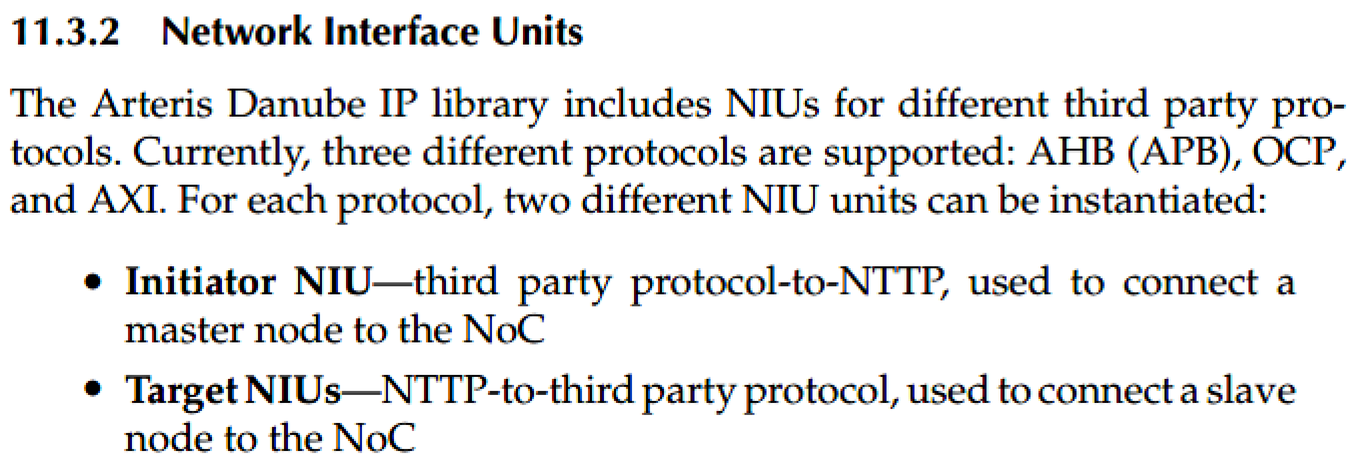
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

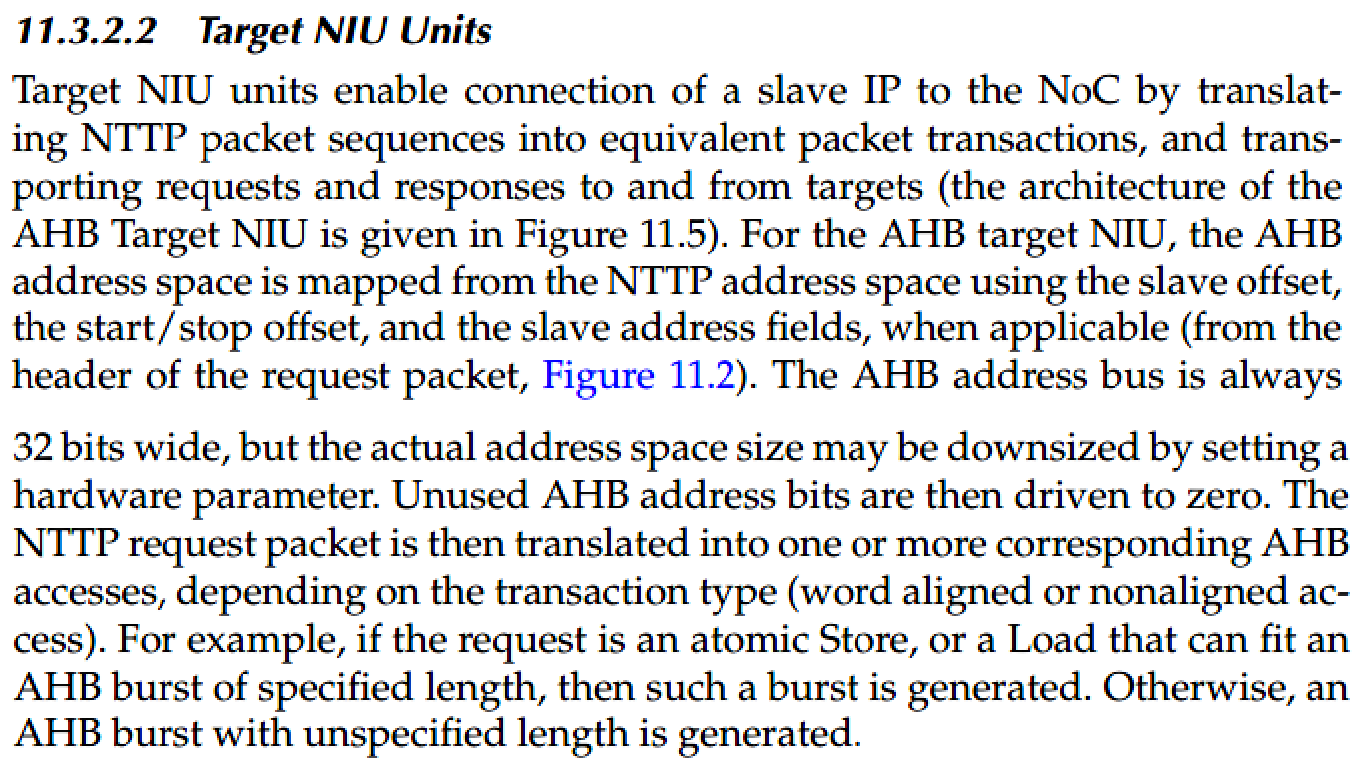
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect;	The Arteris NoC utilized by the Exynos SoC connects the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

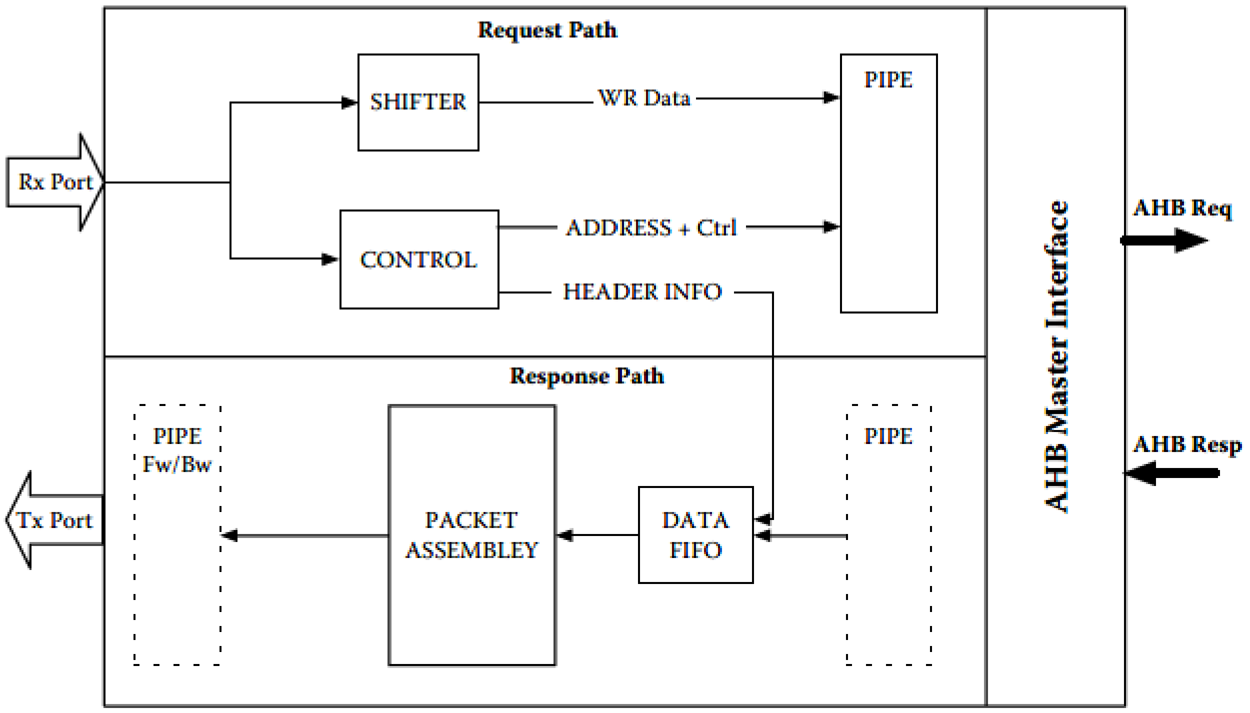
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

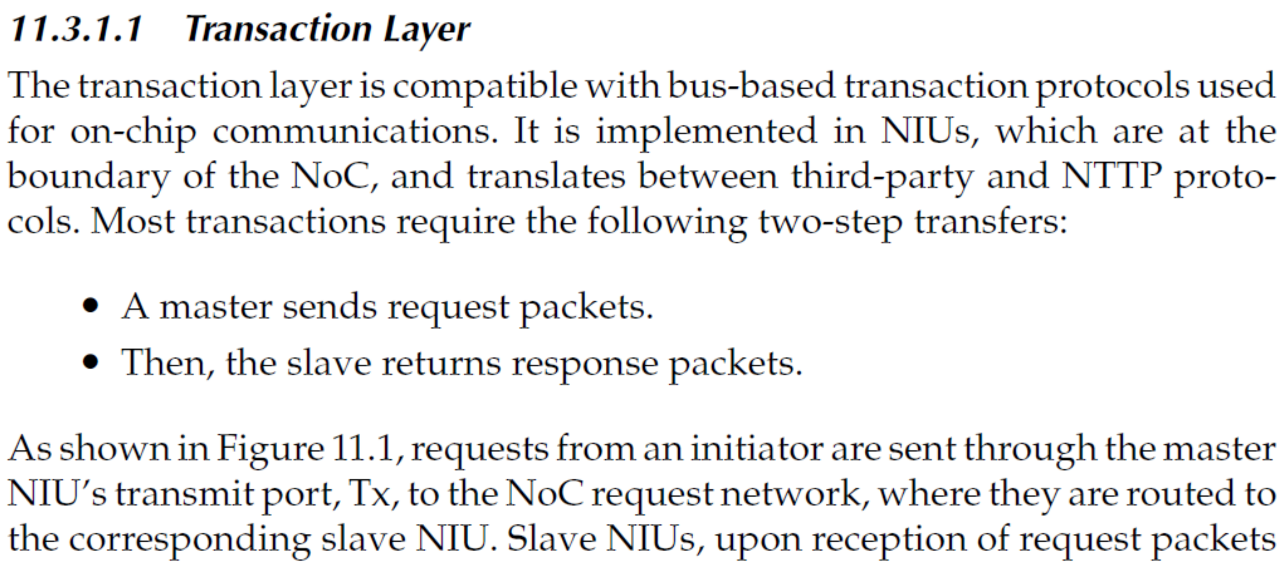
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

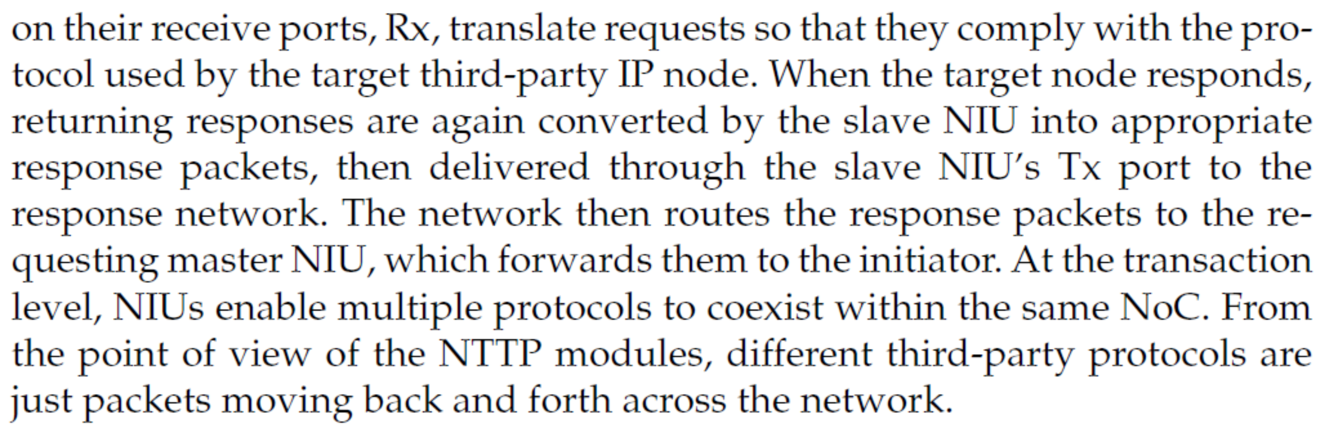
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

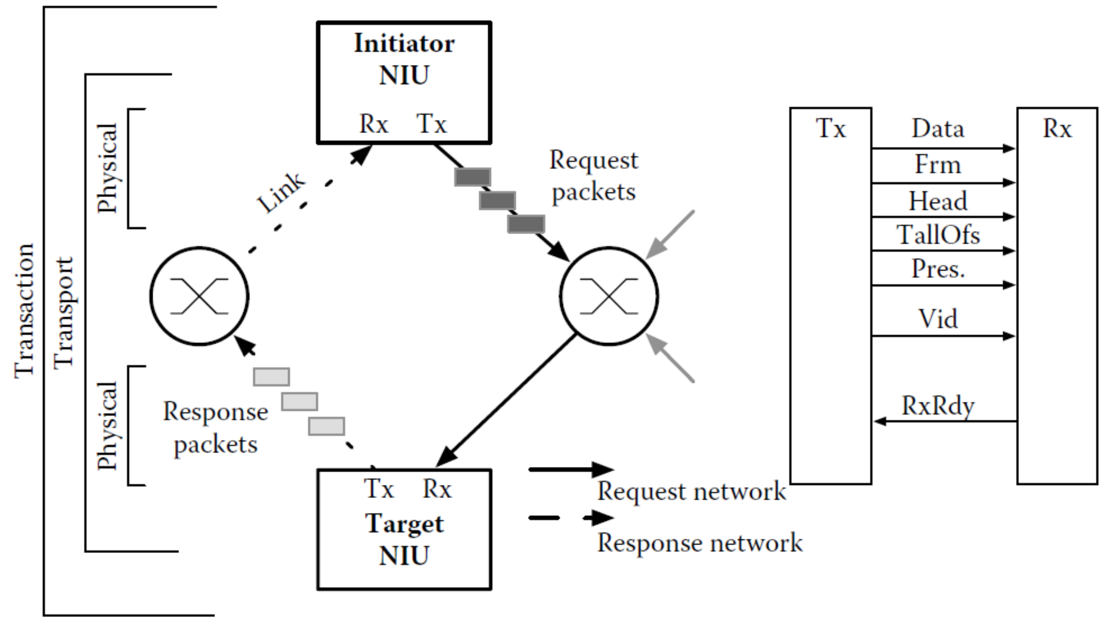
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets":

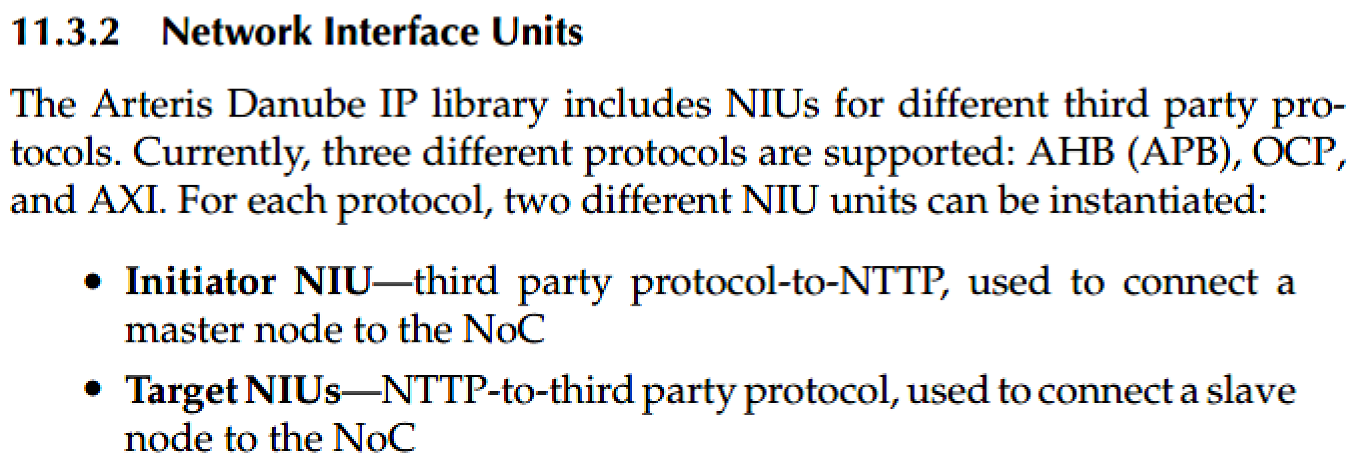
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

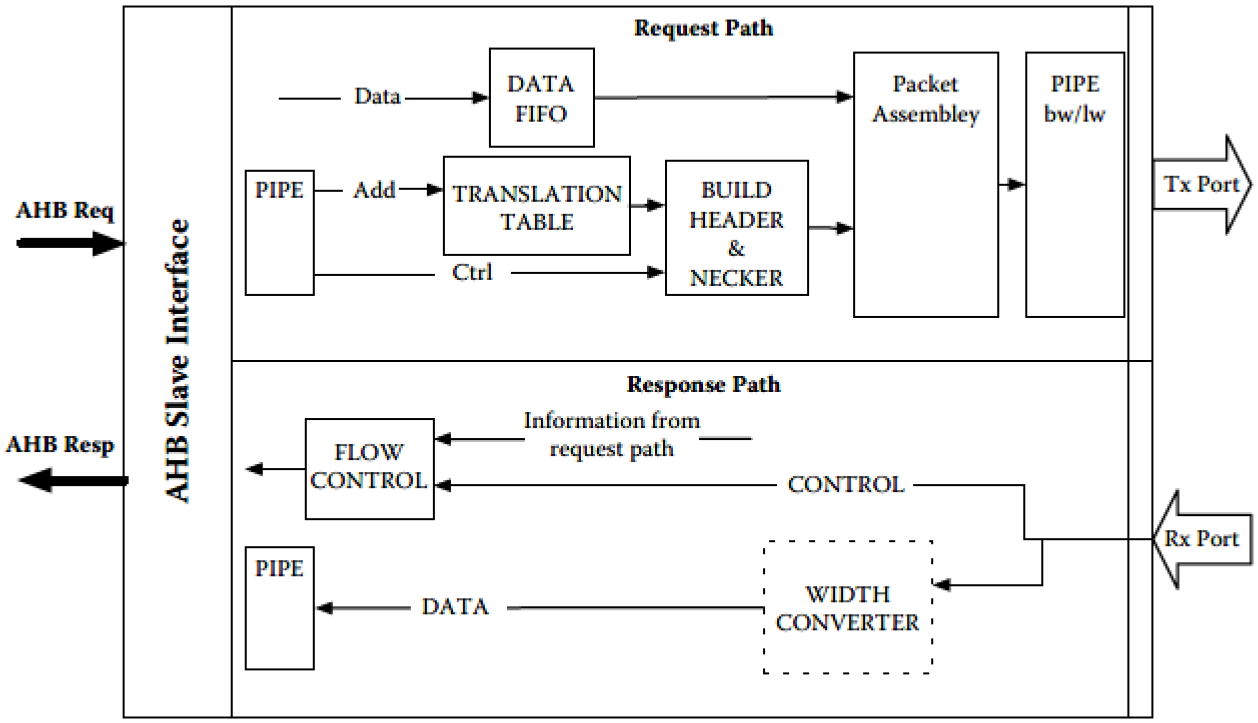
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	Target NIU Architecture  Request Path SHIFTER WR Data PIPE Rx Port CONTROL ADDRESS + Ctrl HEADER INFO AHB Master Interface AHB Req AHB Resp Response Path PIPE Fw/Bw PIPE Tx Port PACKET ASSEMBLEY DATA FIFO **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
determining by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit;	The Arteris NoC utilized by the Exynos SoC determines by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

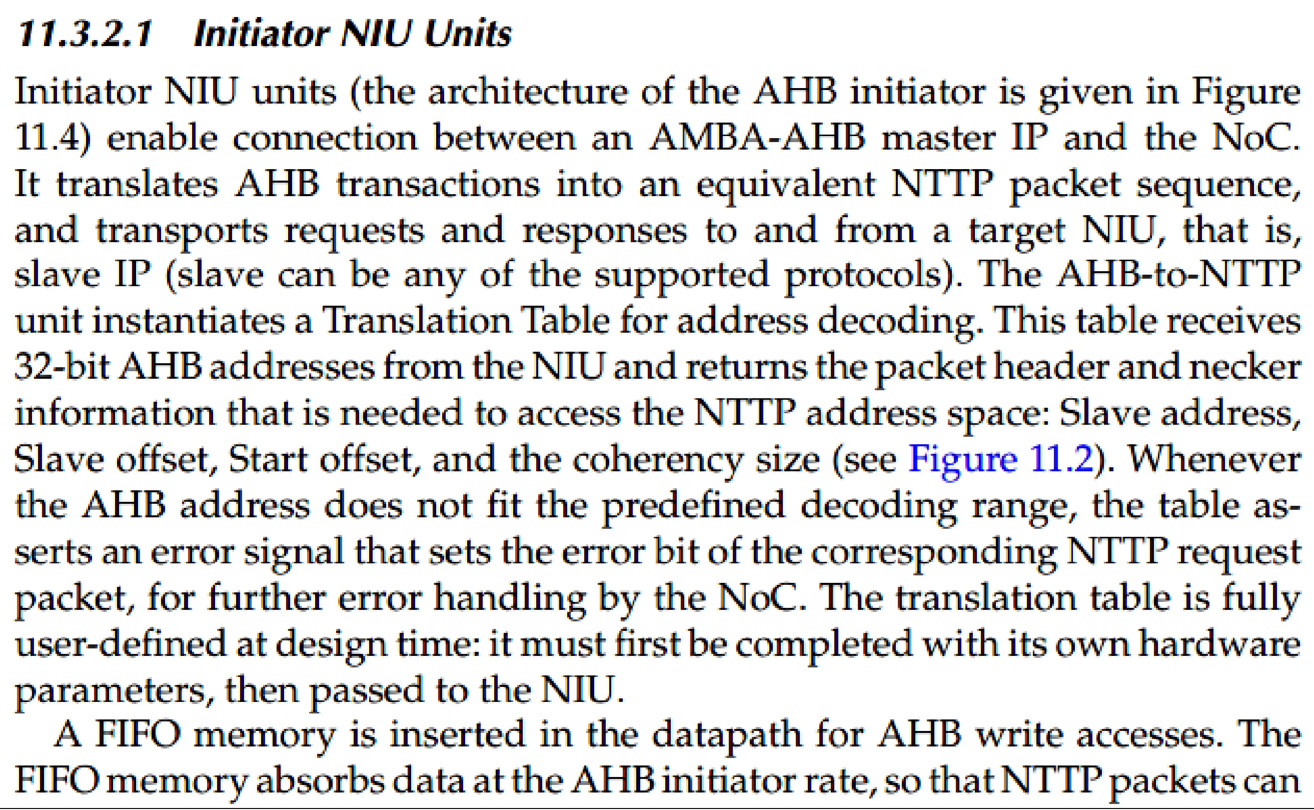
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

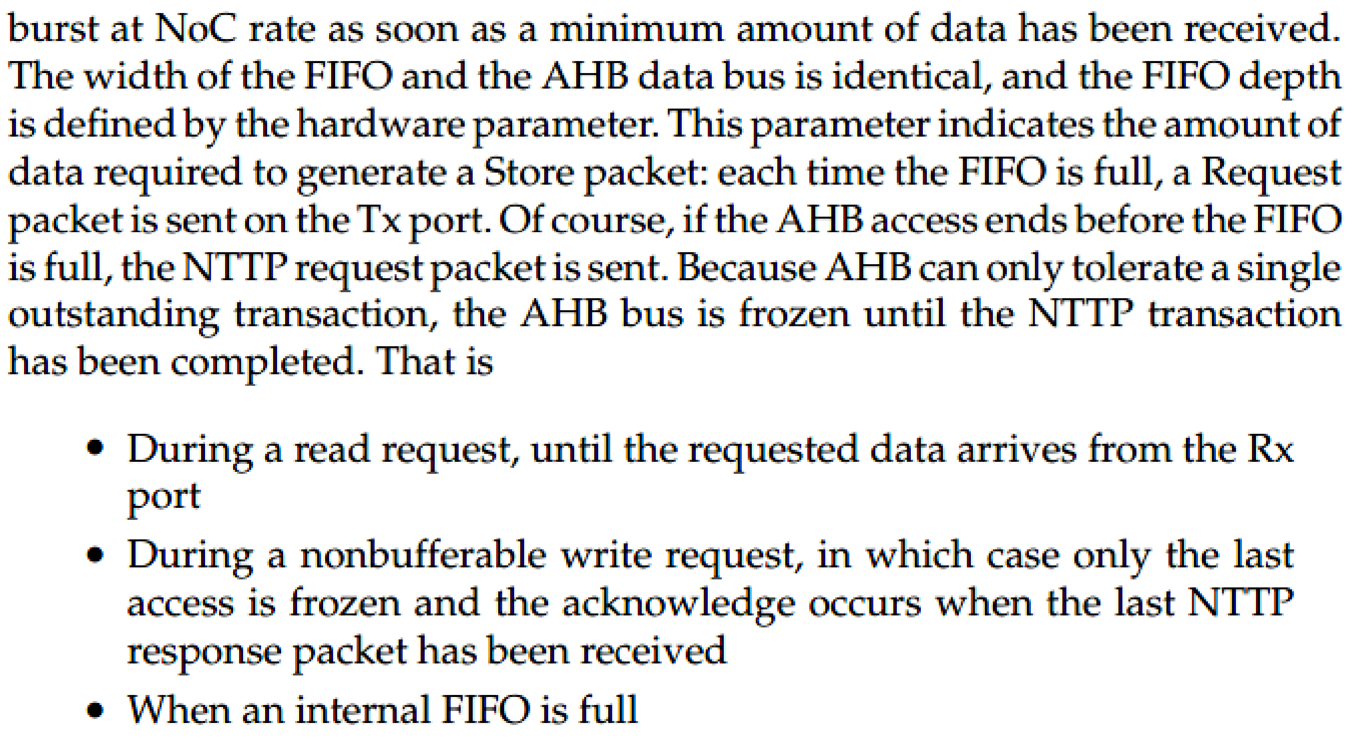
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Transaction Transport Physical Physical Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

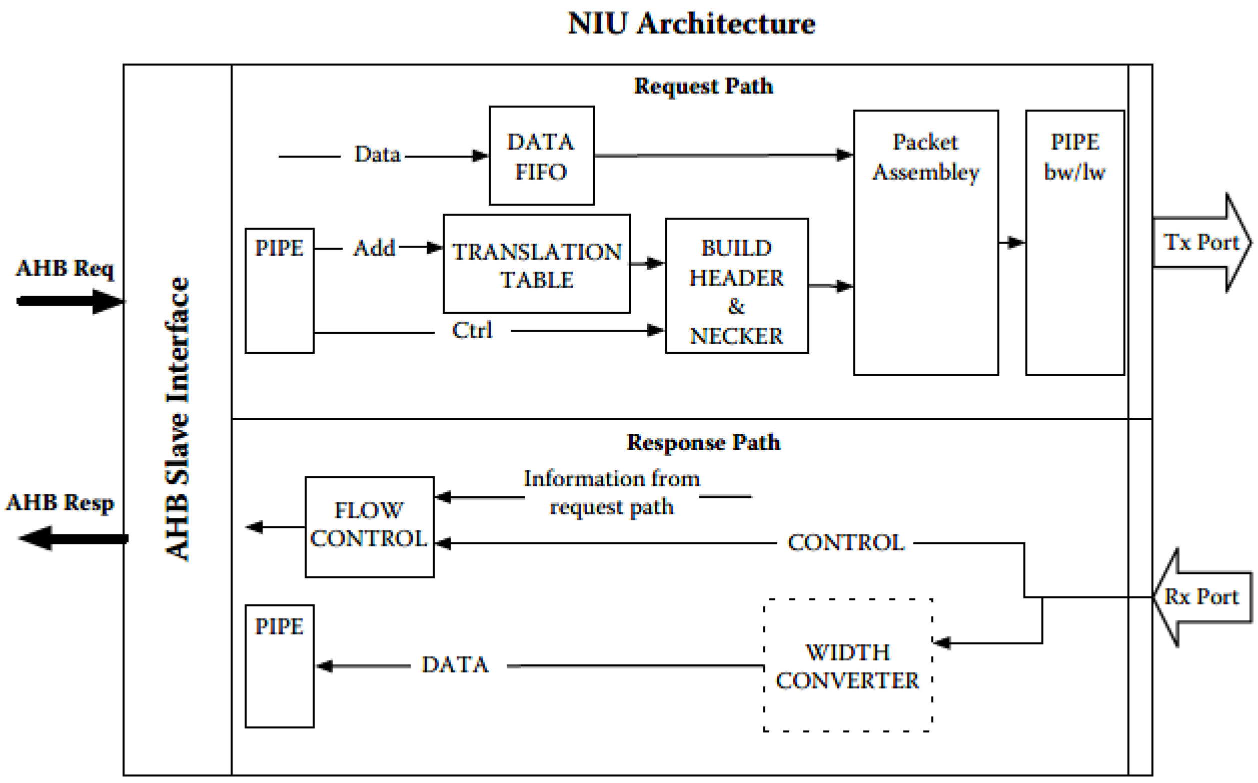
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	NIU Architecture  Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path AHB Resp FLOW CONTROL Information from request path CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

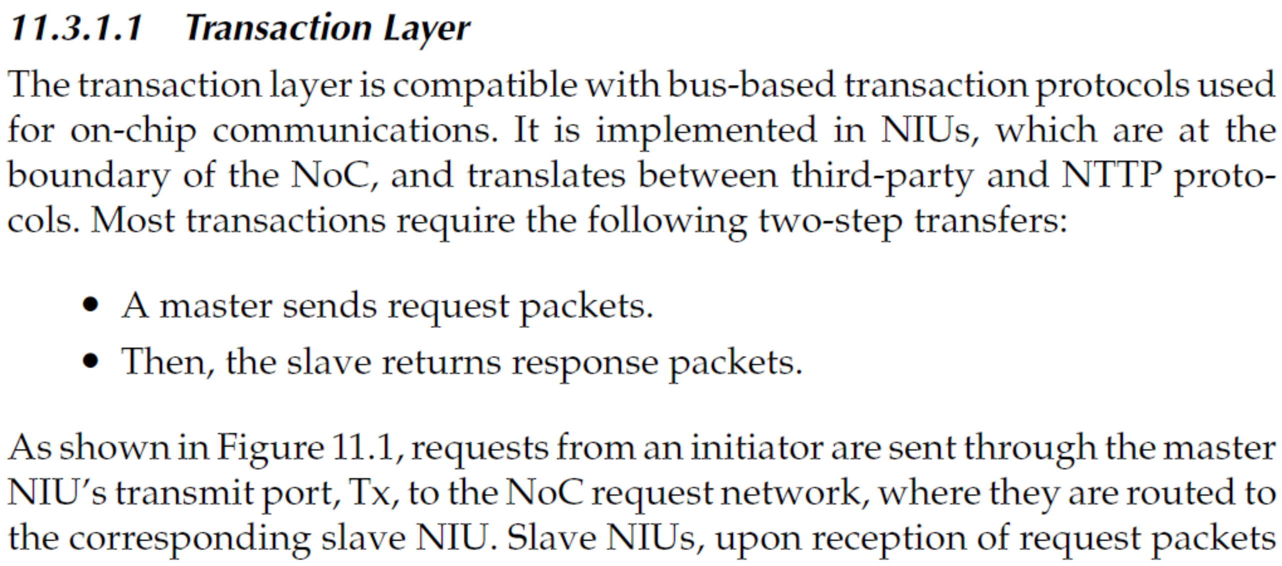
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

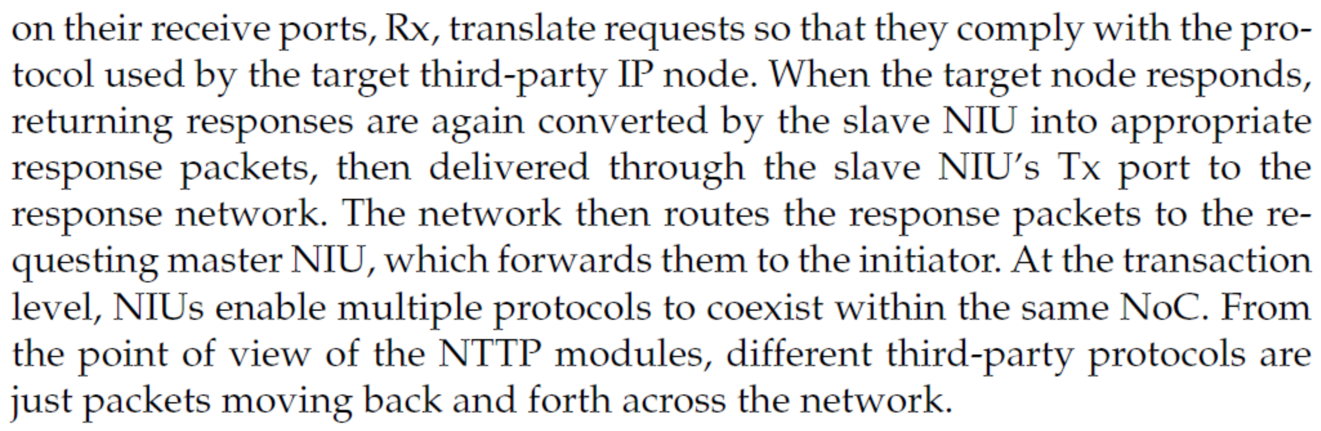
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port":  ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

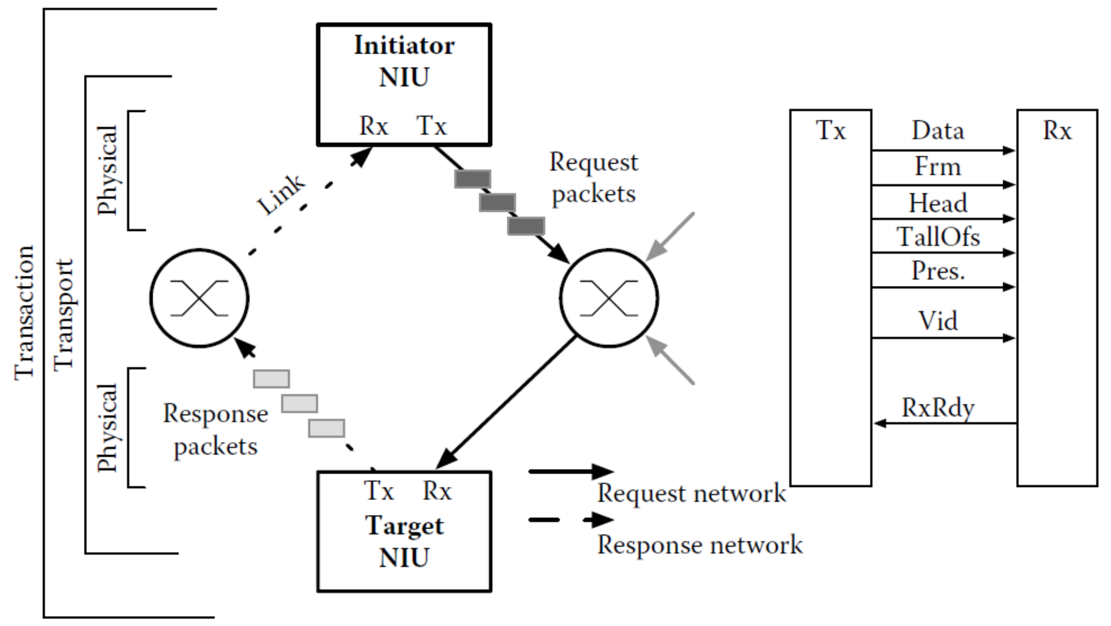
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

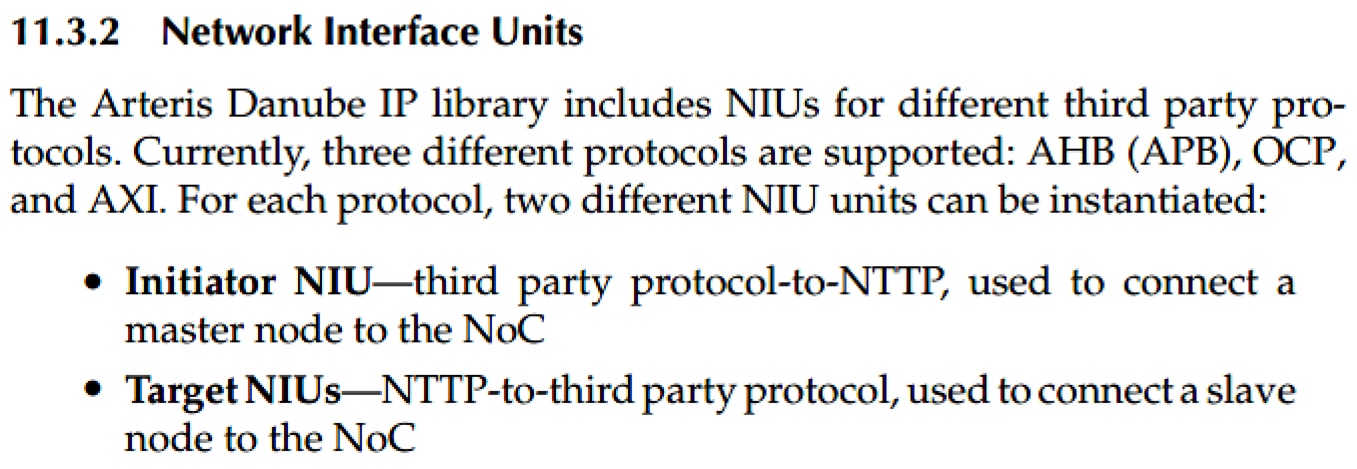
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

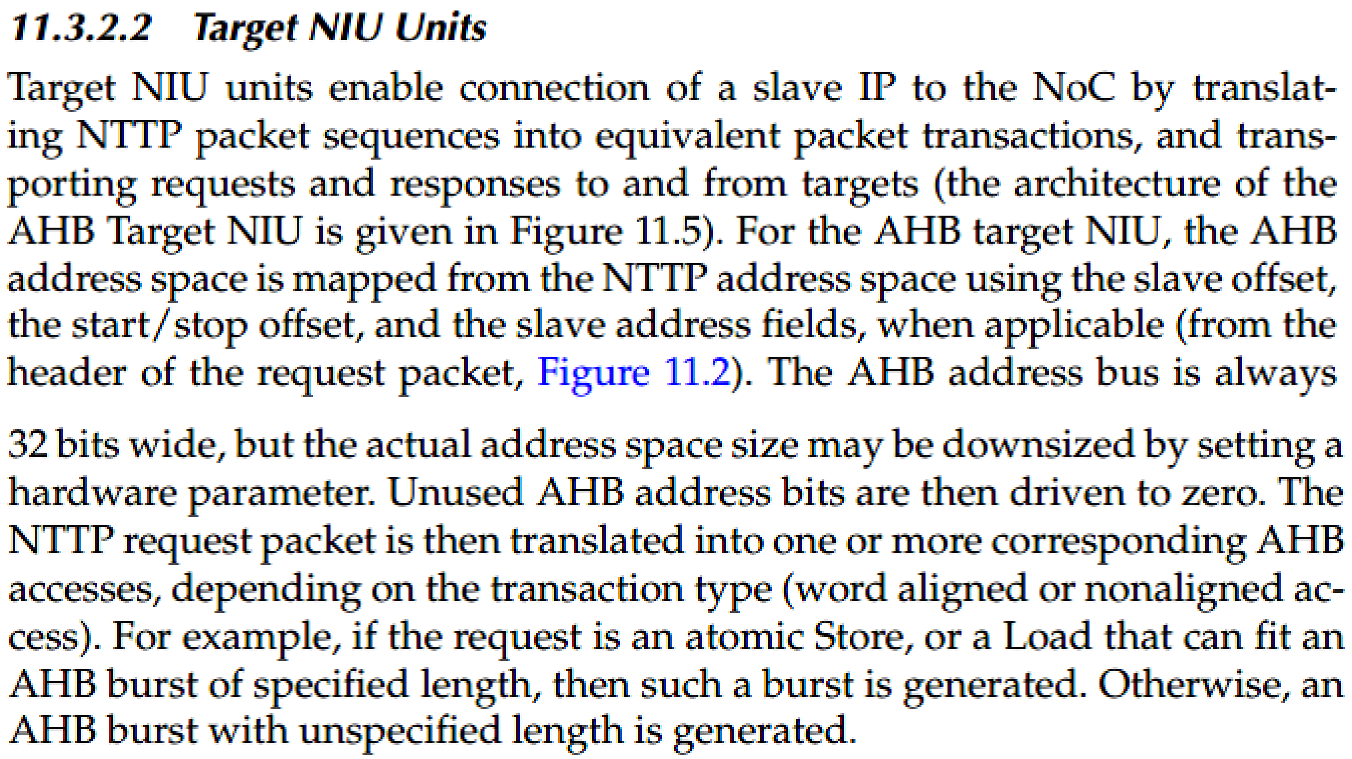
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl AHB Req AHB Slave Interface Tx Port Response Path AHB Resp FLOW CONTROL Information from request path CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
determining by a slave determination unit	The Arteris NoC utilized by the Exynos SoC determines by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit, either literally or under the doctrine of equivalents.

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

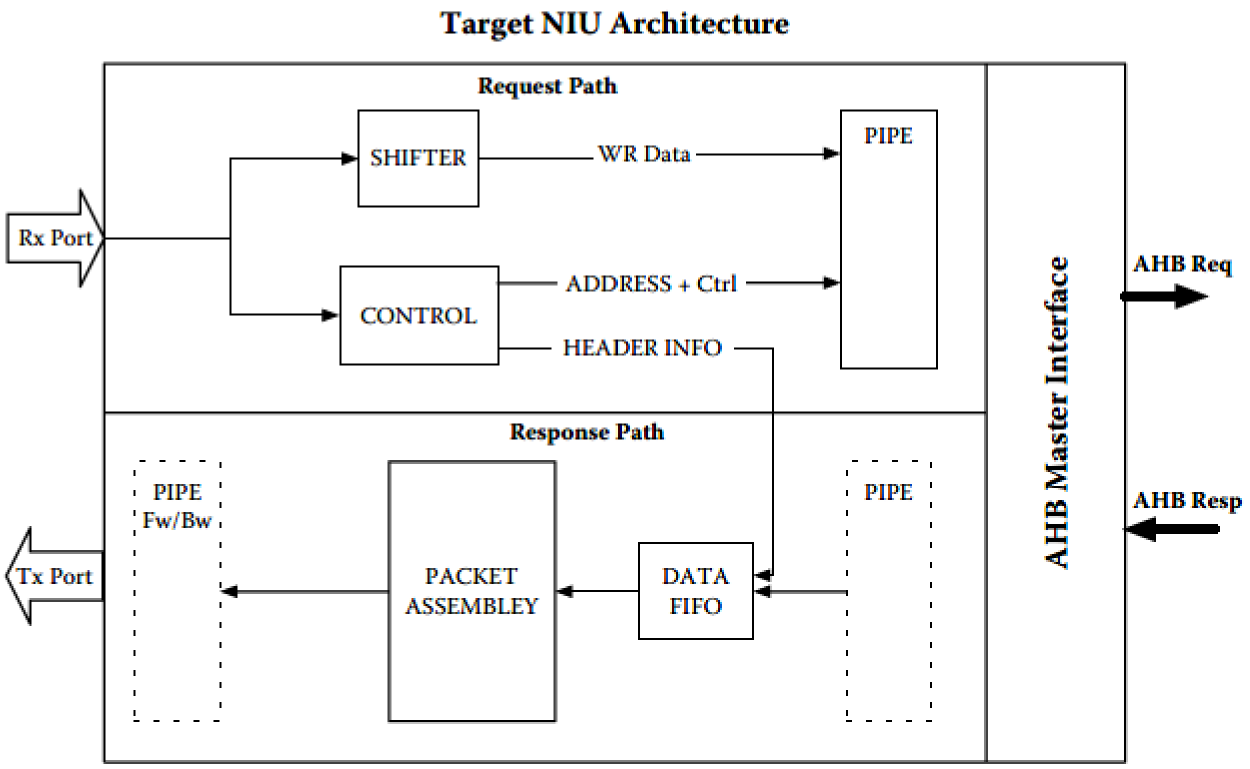
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

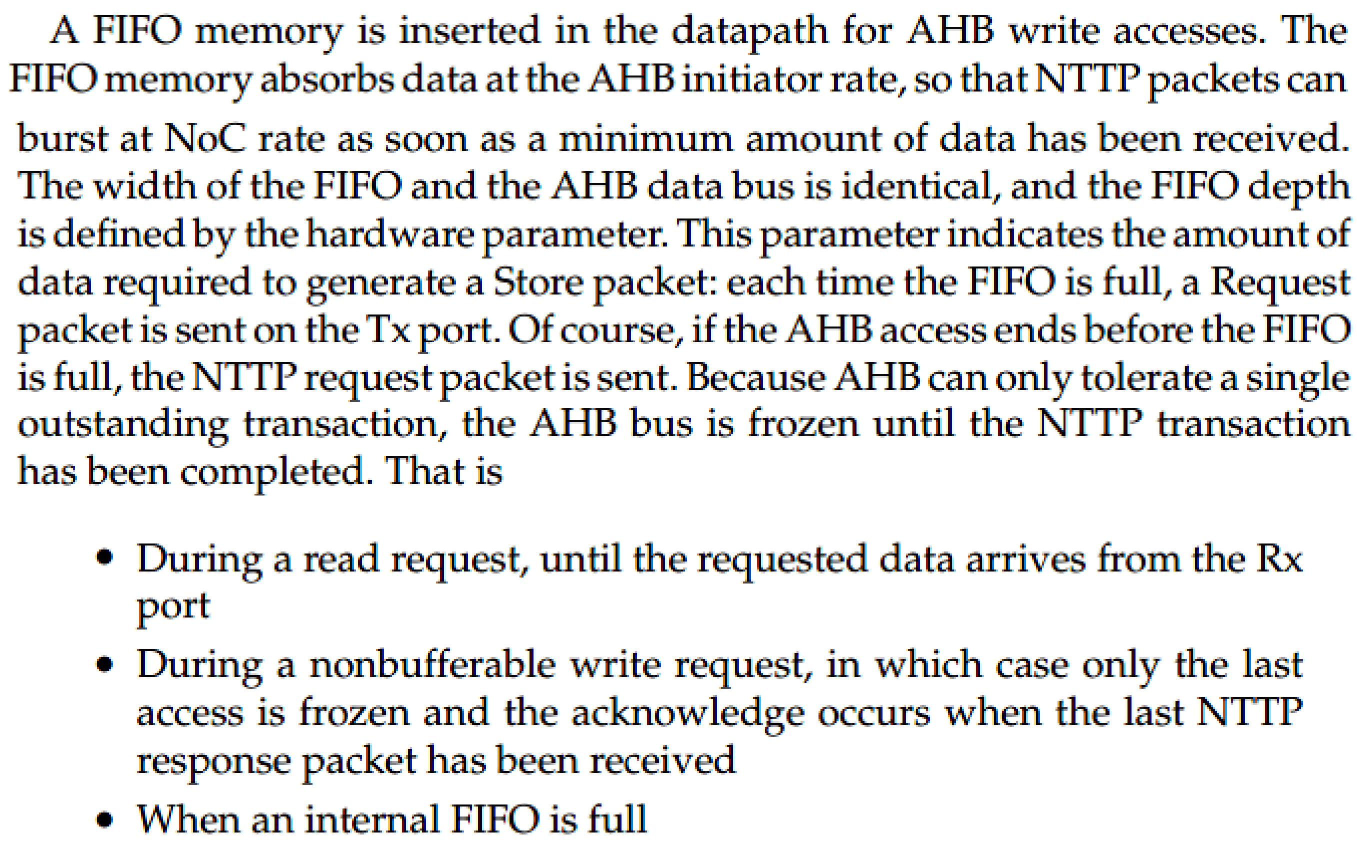
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

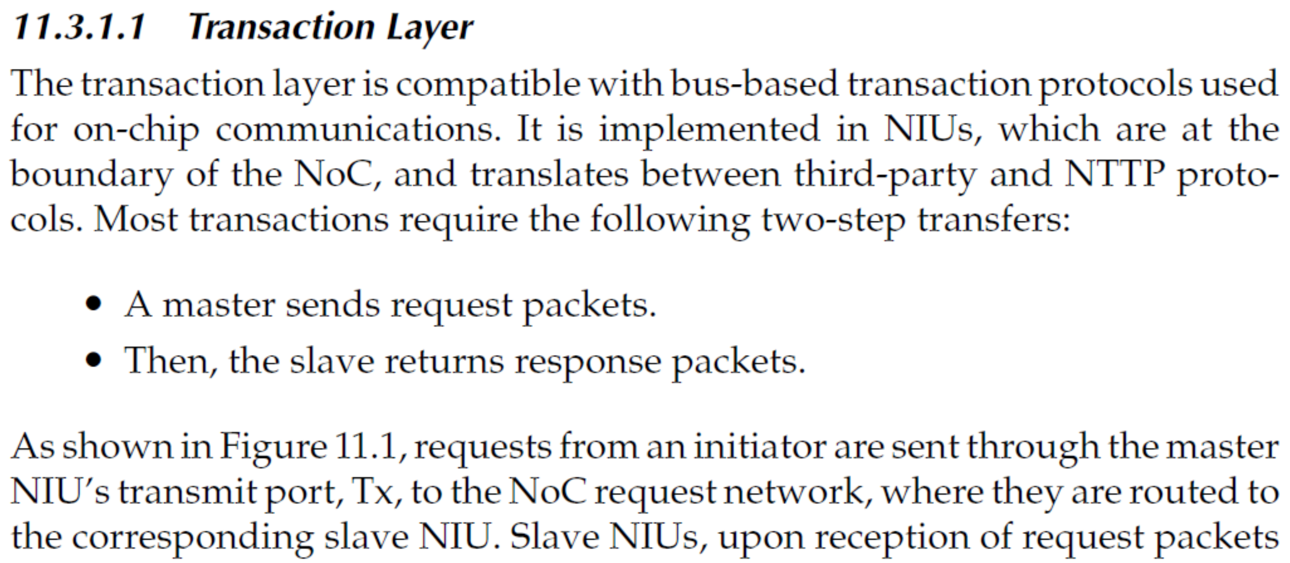
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Target NIU Architecture Request Path SHIFTER WR Data PIPE Rx Port ADDRESS + Ctrl CONTROL HEADER INFO AHB Master Interface AHB Req AHB Resp Response Path PIPE Fw/Bw PACKET ASSEMBLEY DATA FIFO PIPE Tx Port **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

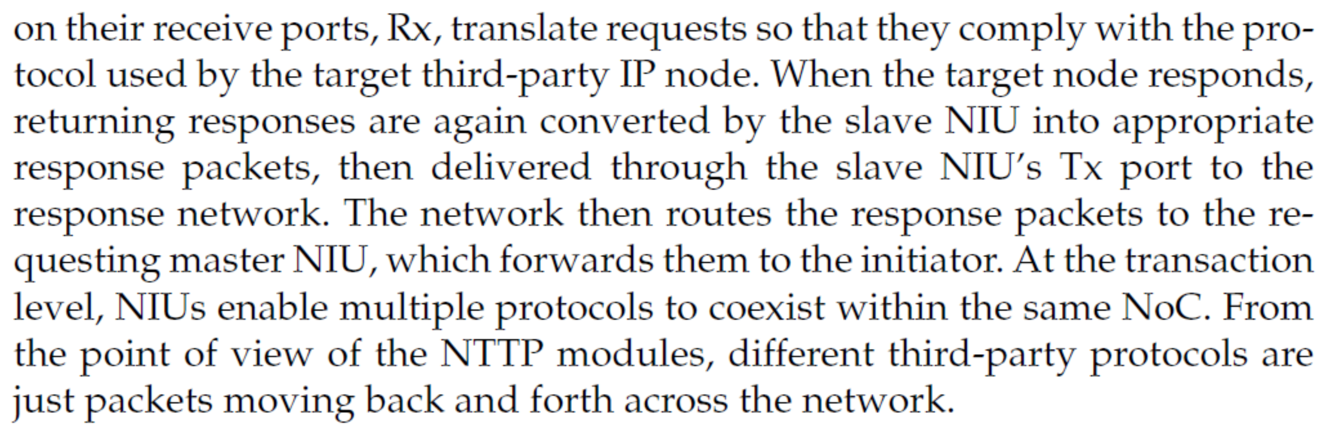
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

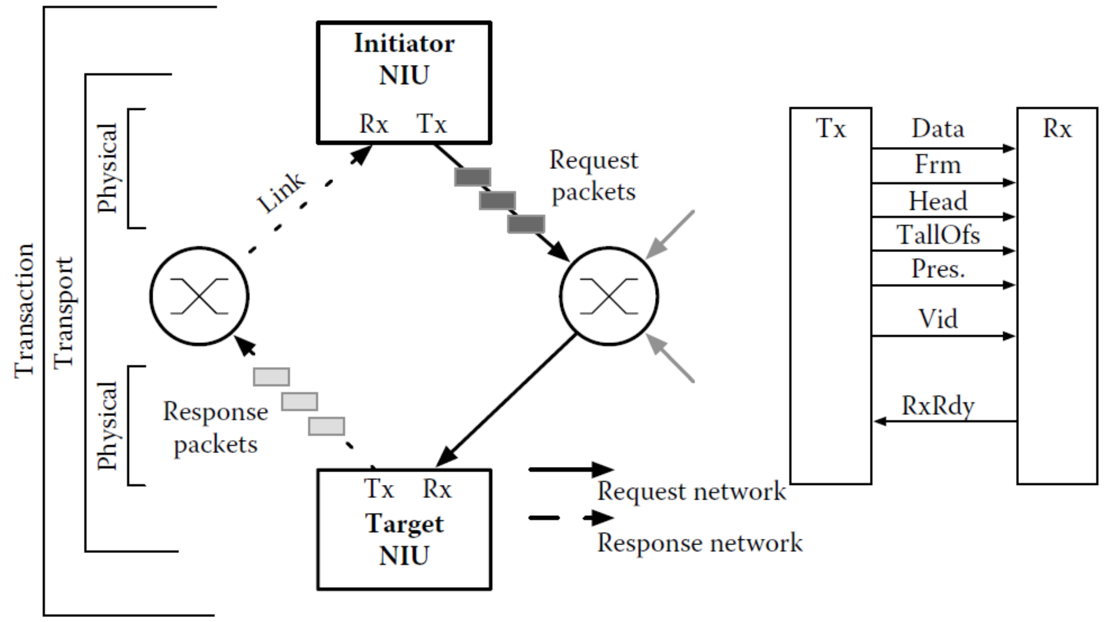
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port":  A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

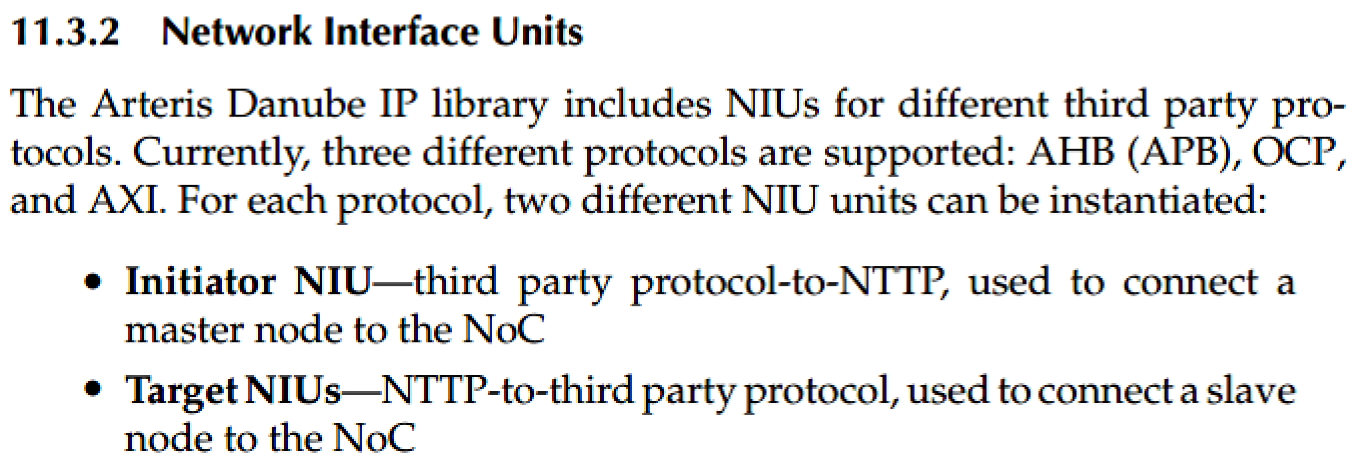
4855-4585-1715.1

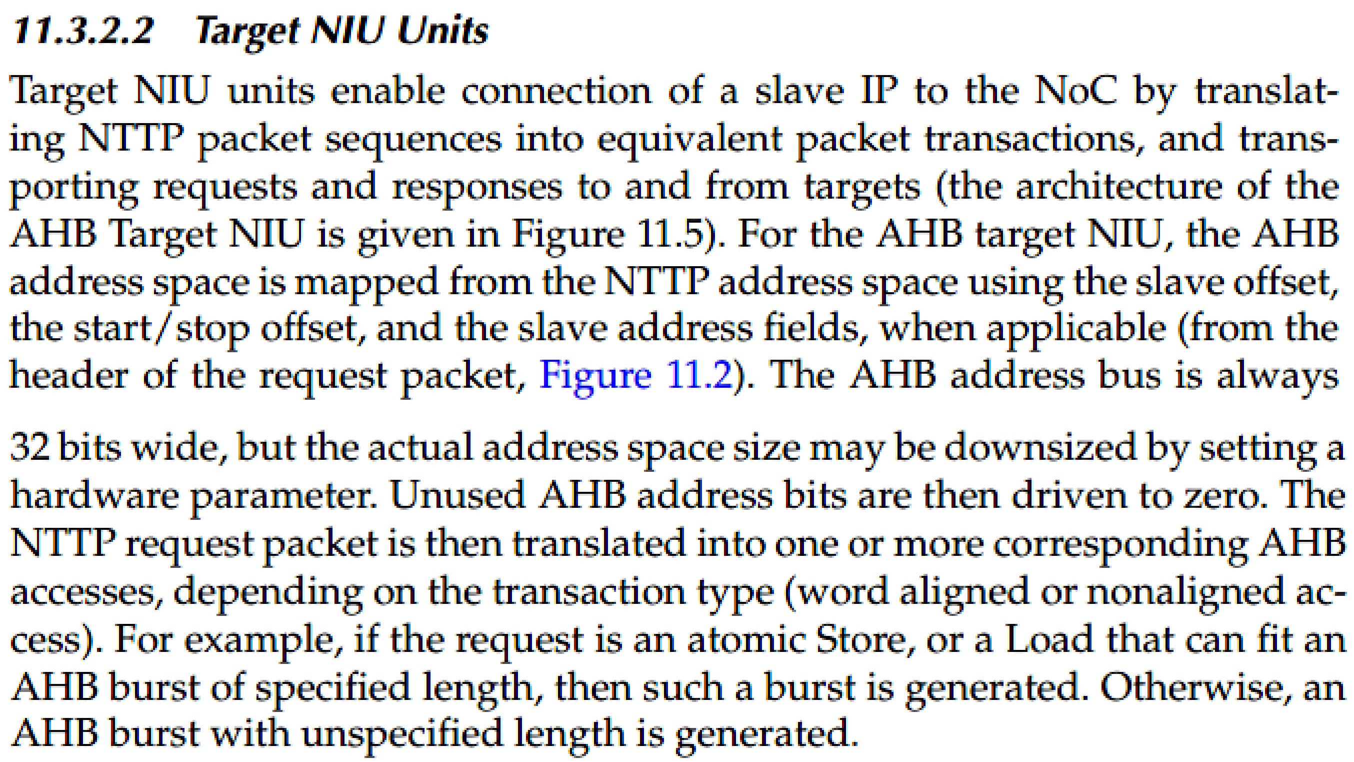
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered;	The Arteris NoC utilized by the Exynos SoC buffers by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

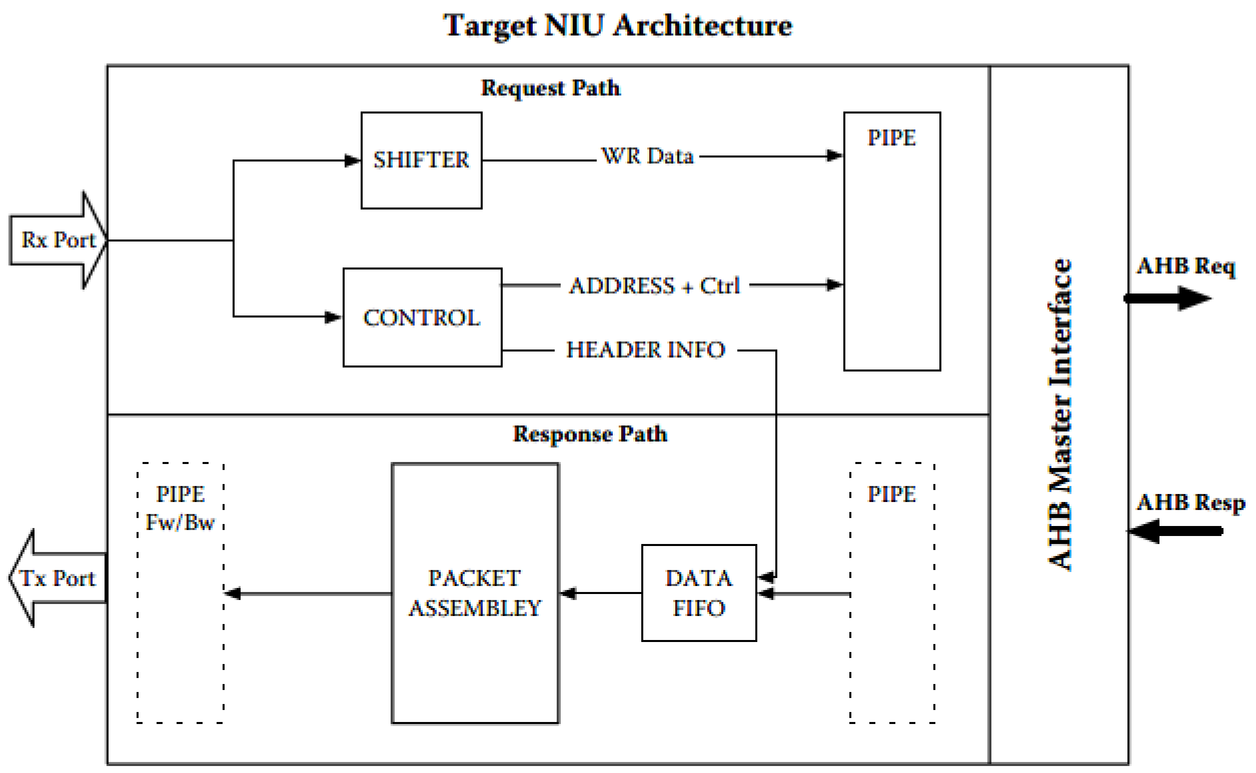
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Initiator NIU Rx Tx Link Request packets Transaction Transport Physical Physical Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

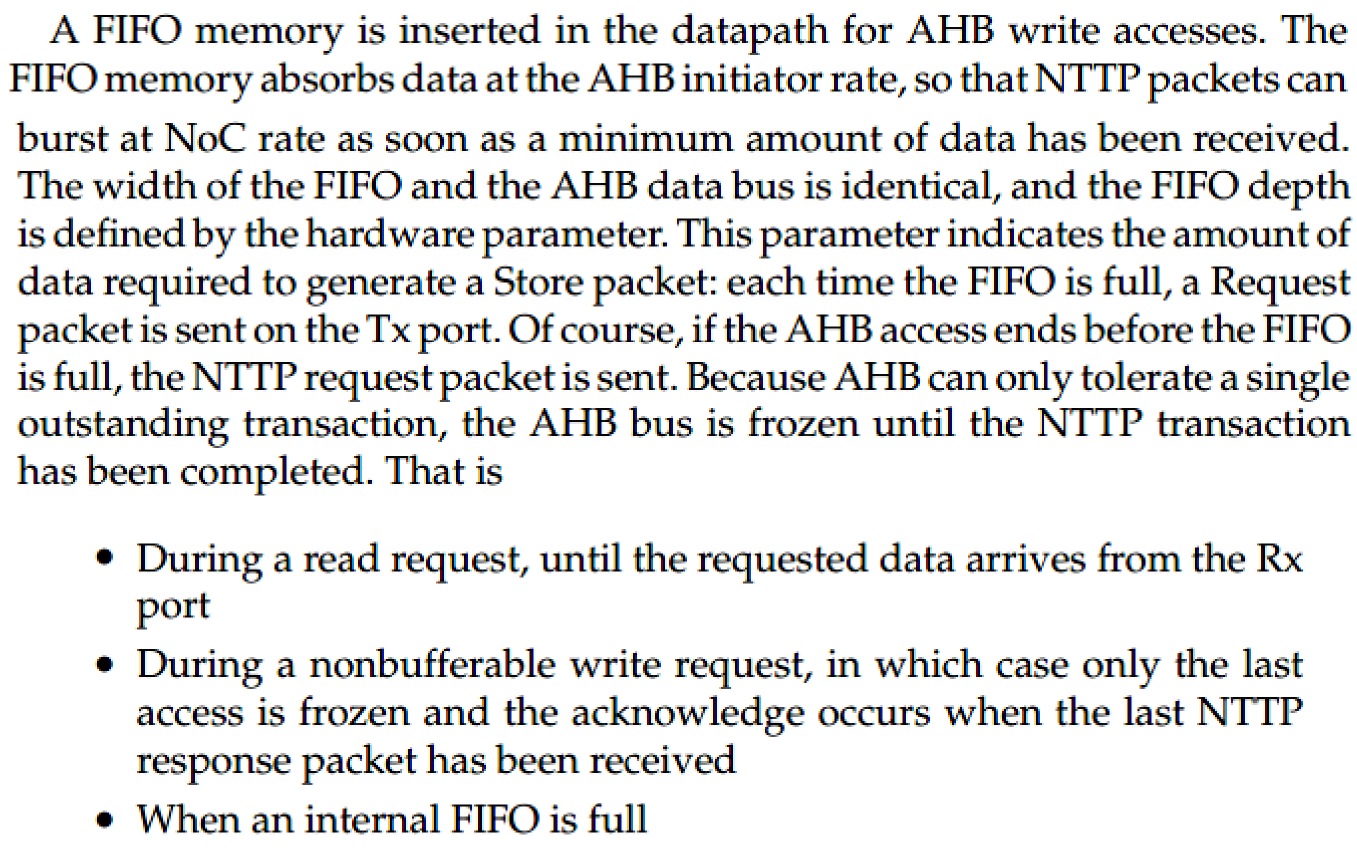
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

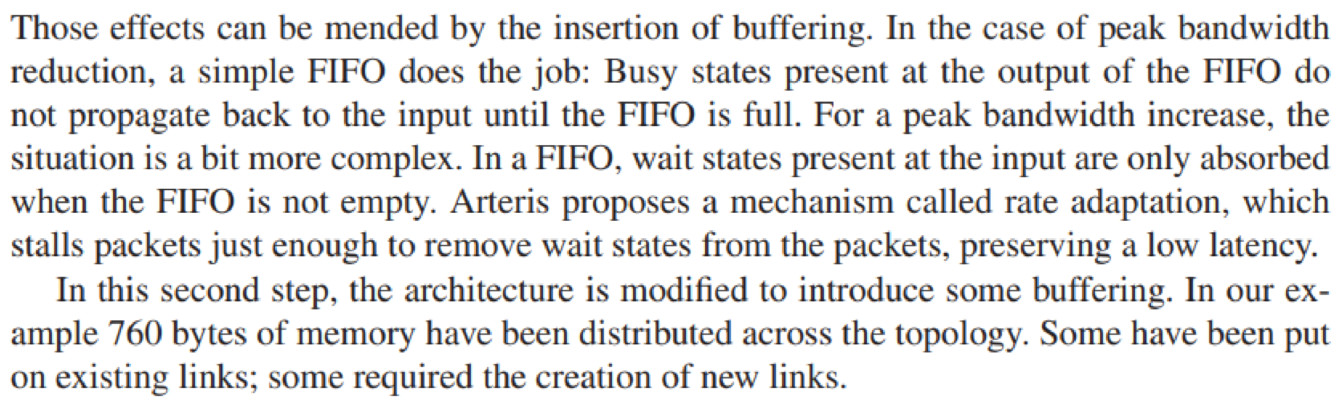
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Target NIU Architecture Request Path SHIFTER WR Data PIPE Rx Port CONTROL ADDRESS + Ctrl HEADER INFO AHB Master Interface AHB Req AHB Resp Response Path PIPE Fw/Bw PIPE Tx Port PACKET ASSEMBLEY DATA FIFO **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

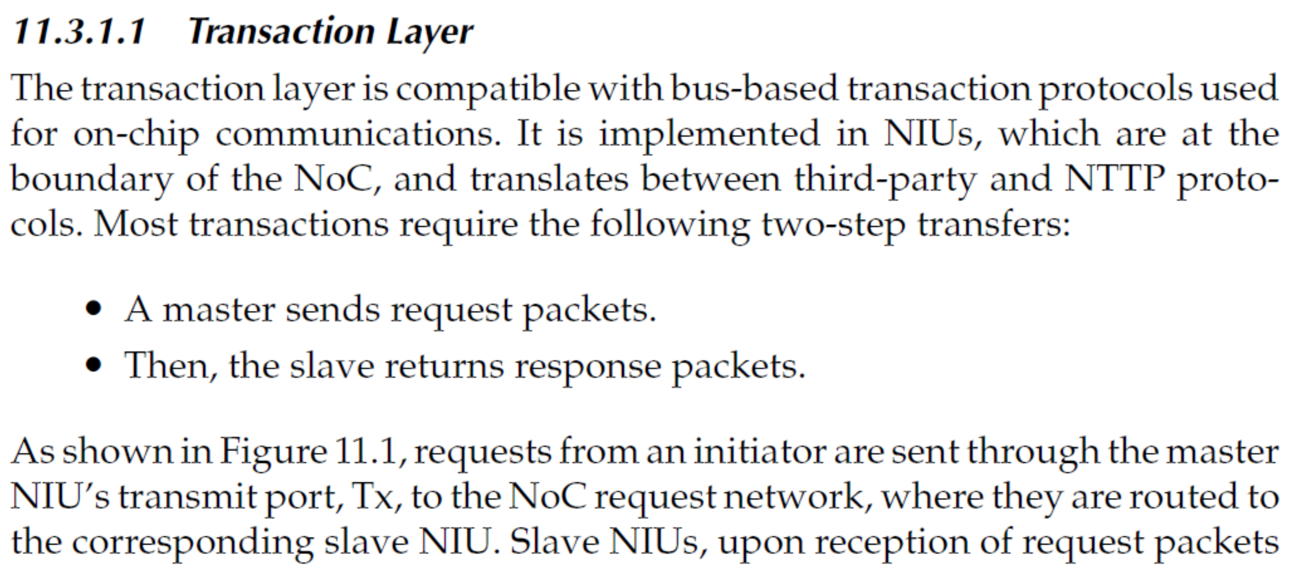
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

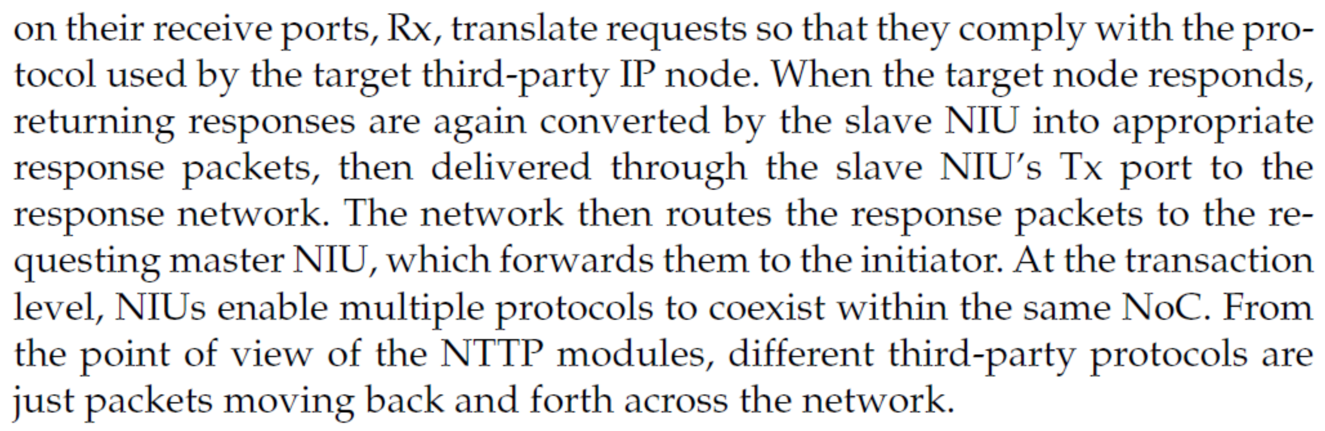
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port":  A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

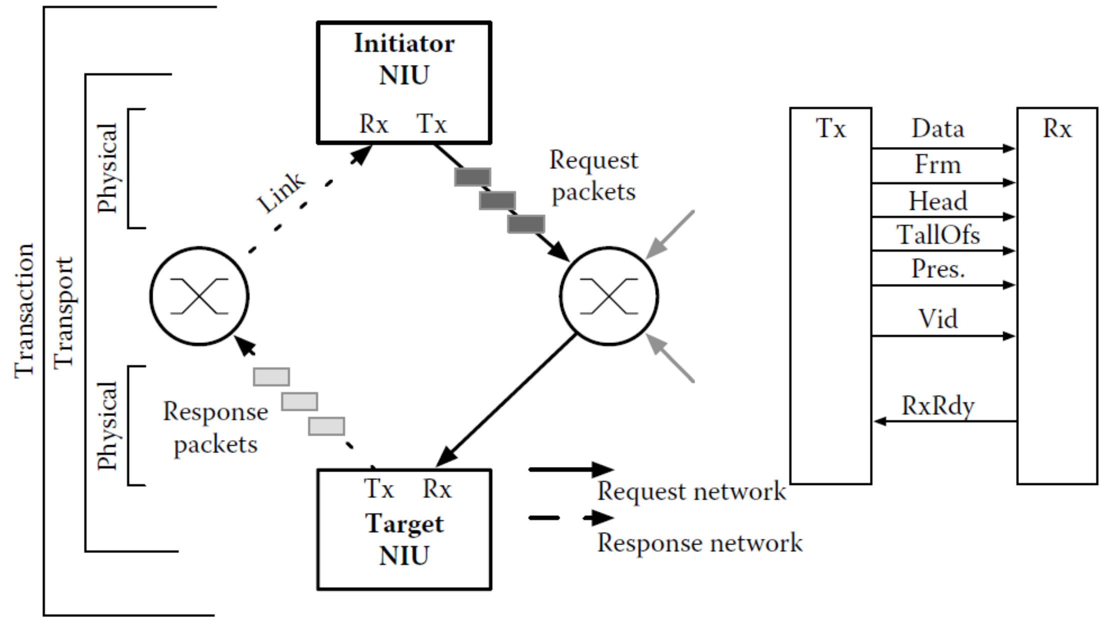
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

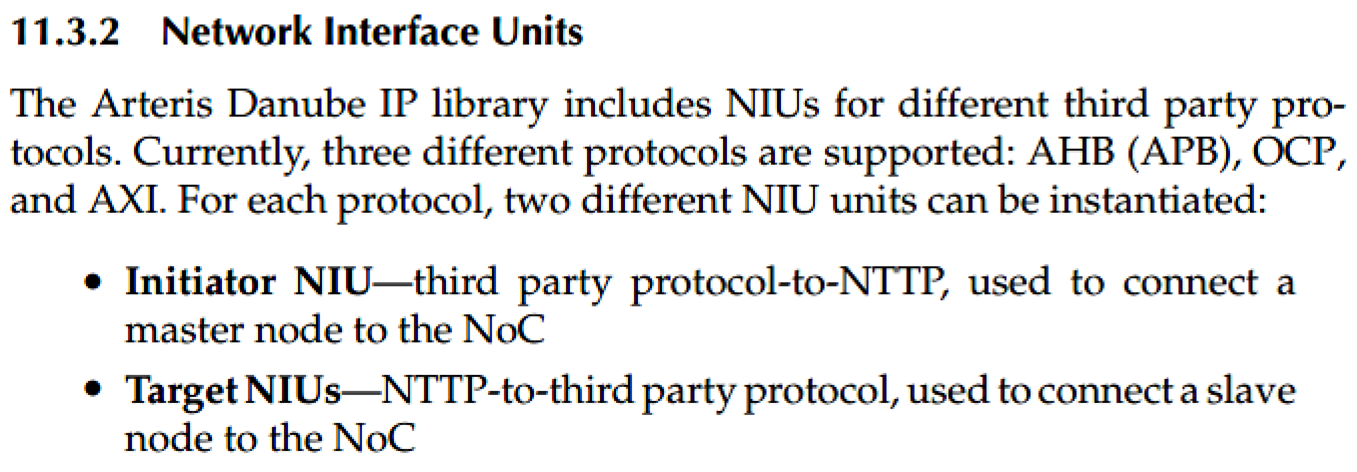
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level":  Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.
transferring the buffered data from the slave wrapper to the master	The Arteris NoC utilized by the Exynos SoC transfers the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

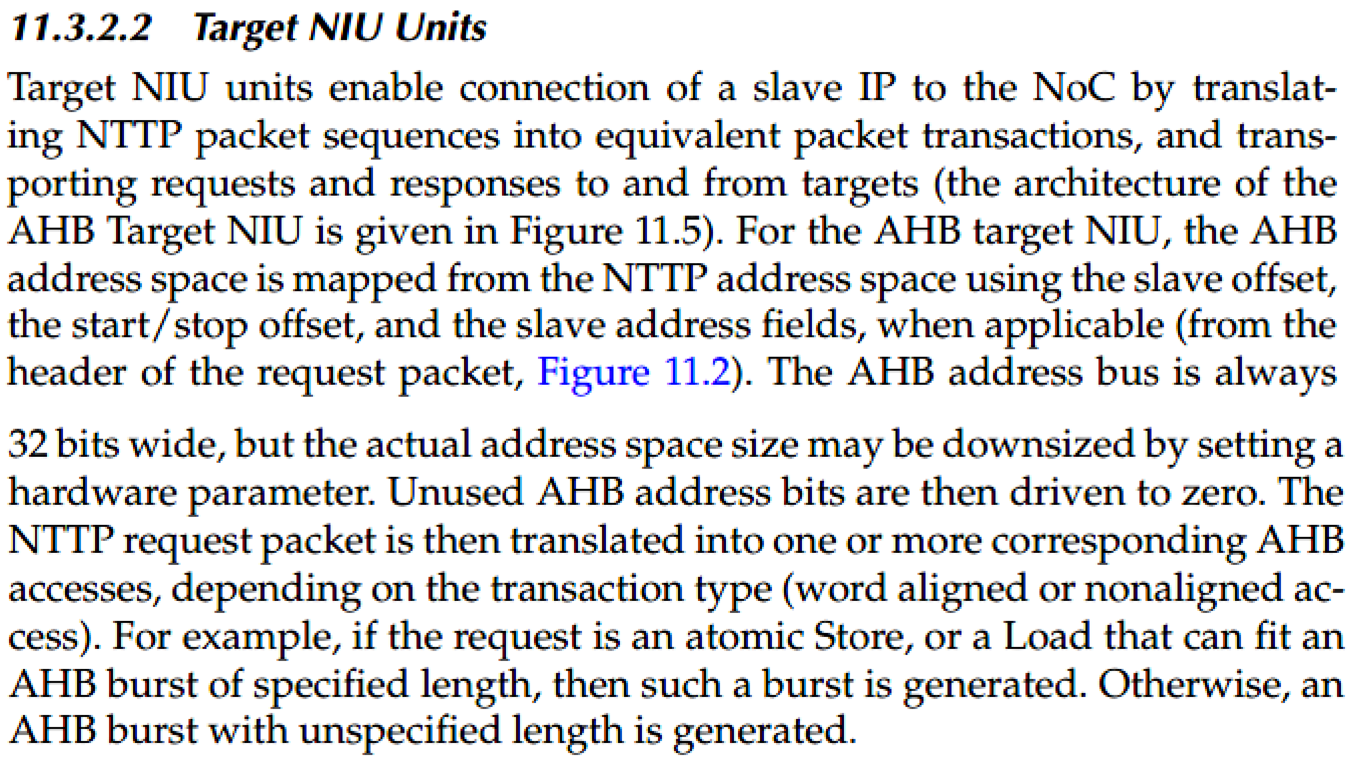
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
wrapper when said first optimal amount of data has been buffered by the slave wrapper;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

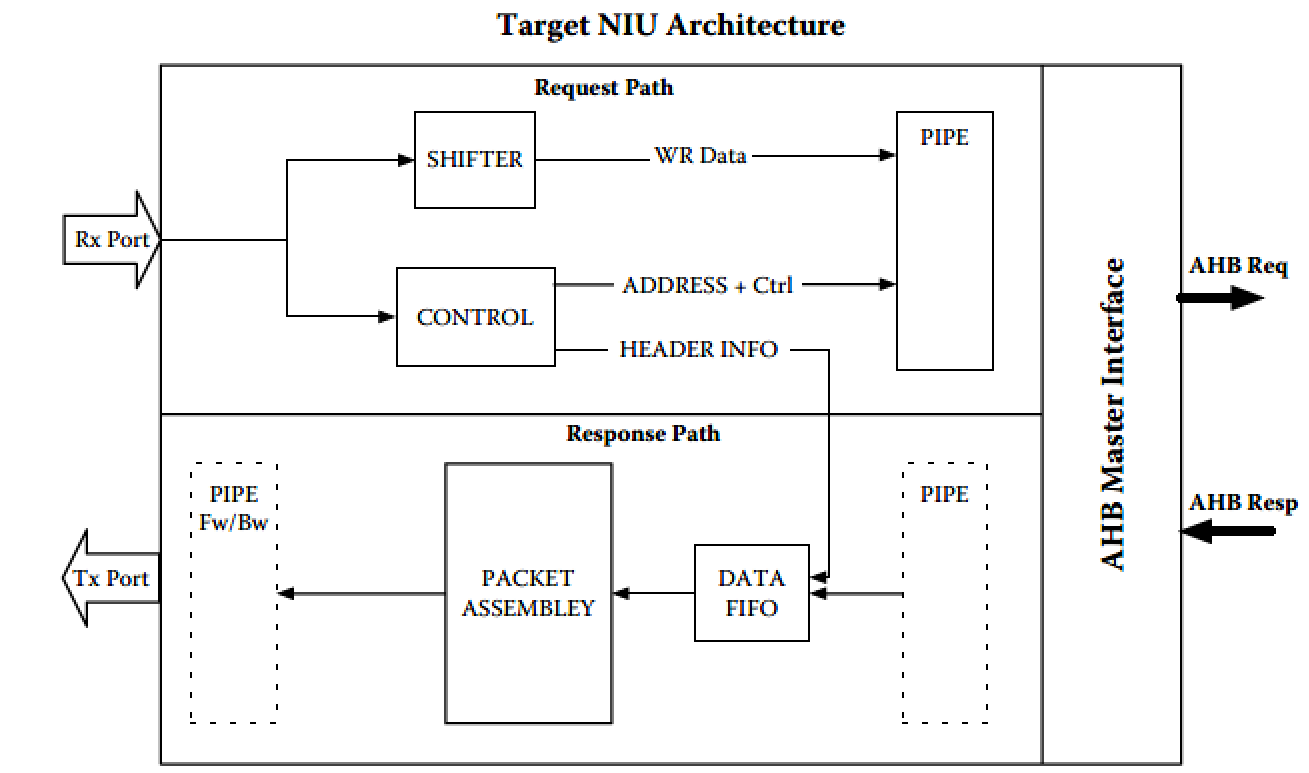
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Transaction Transport Physical Physical Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

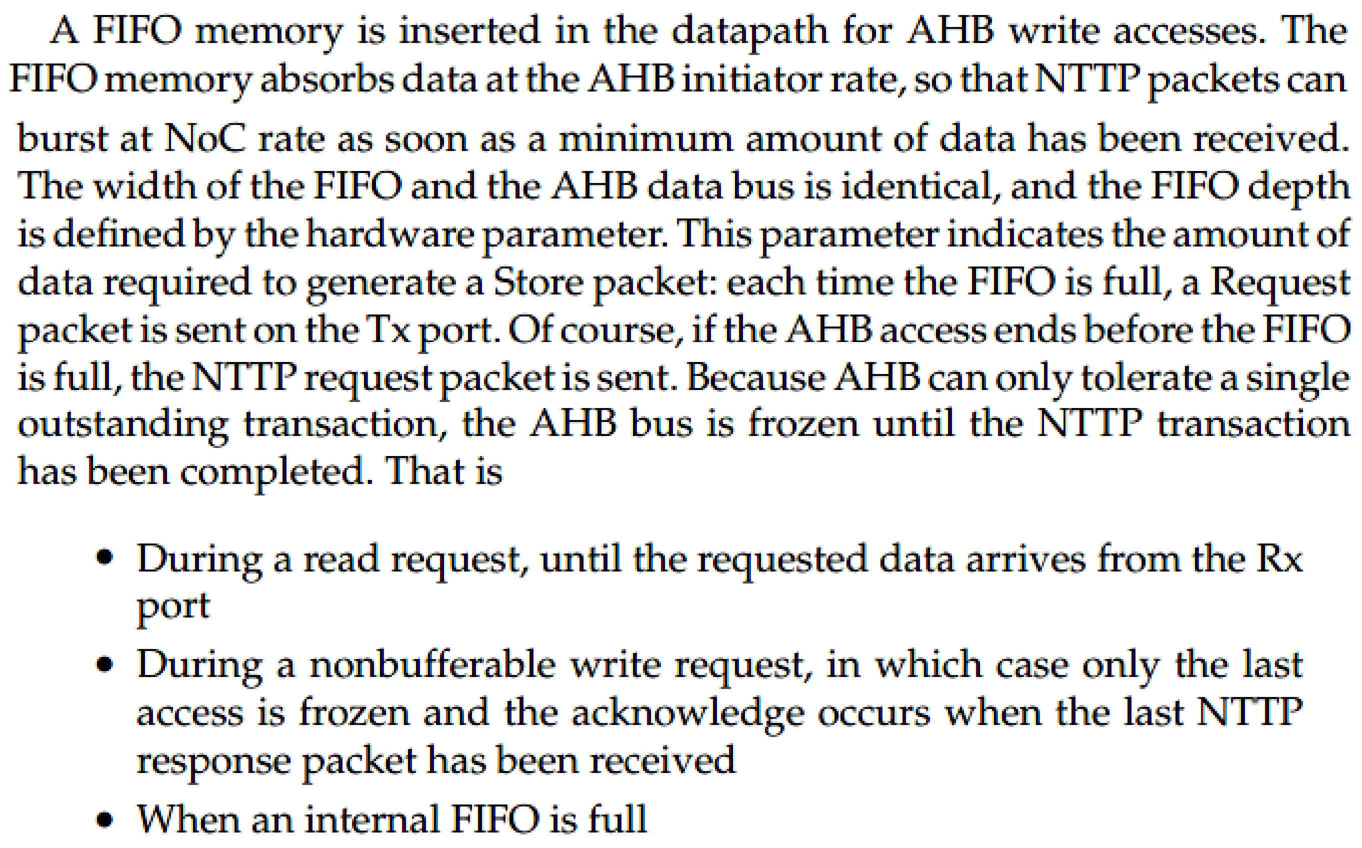
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

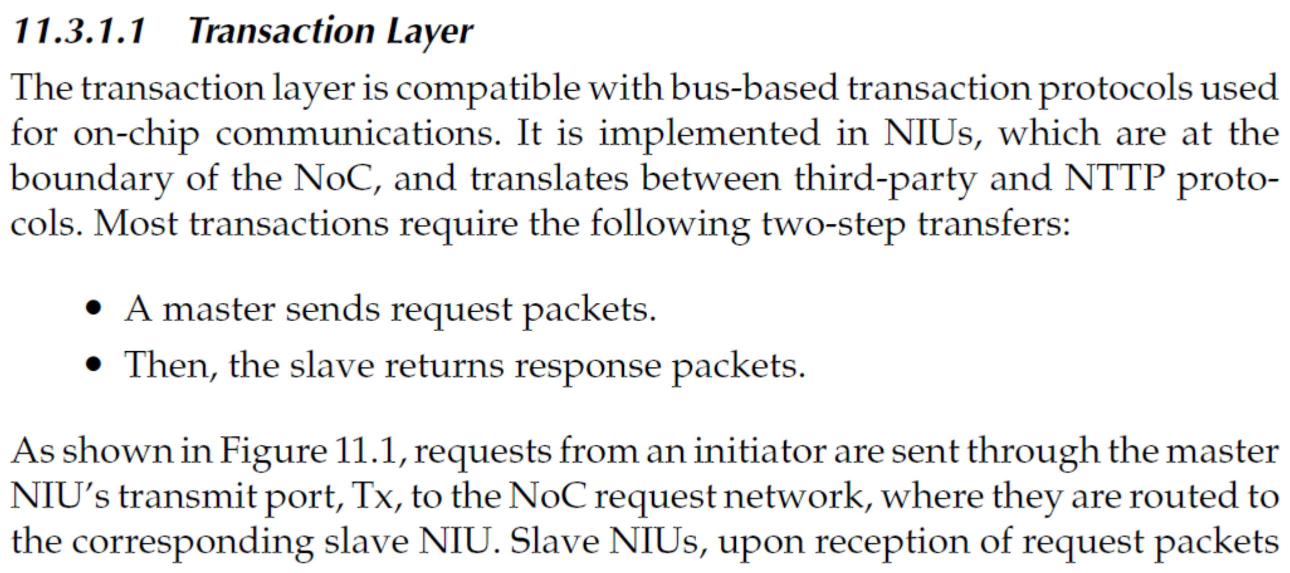
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

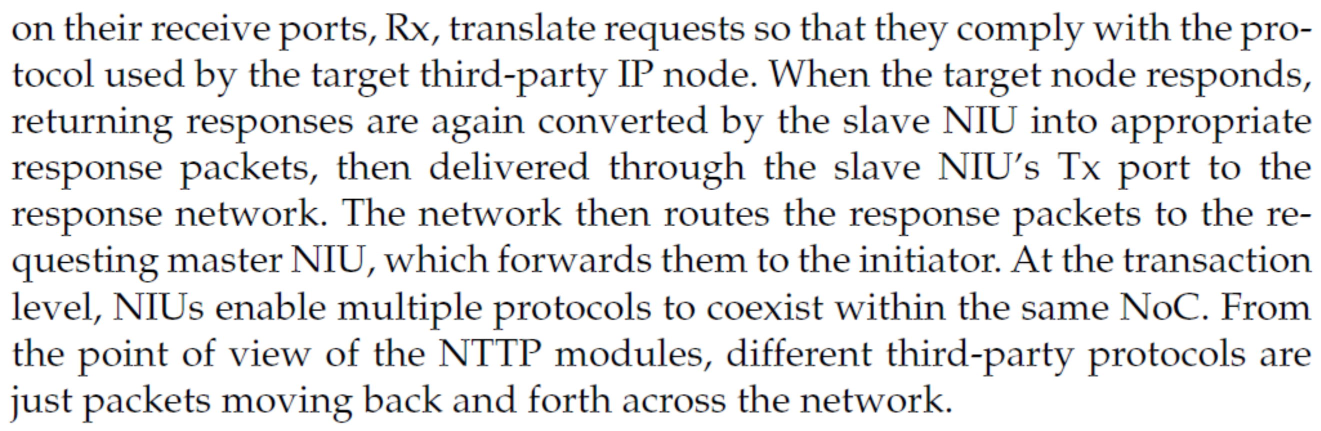
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

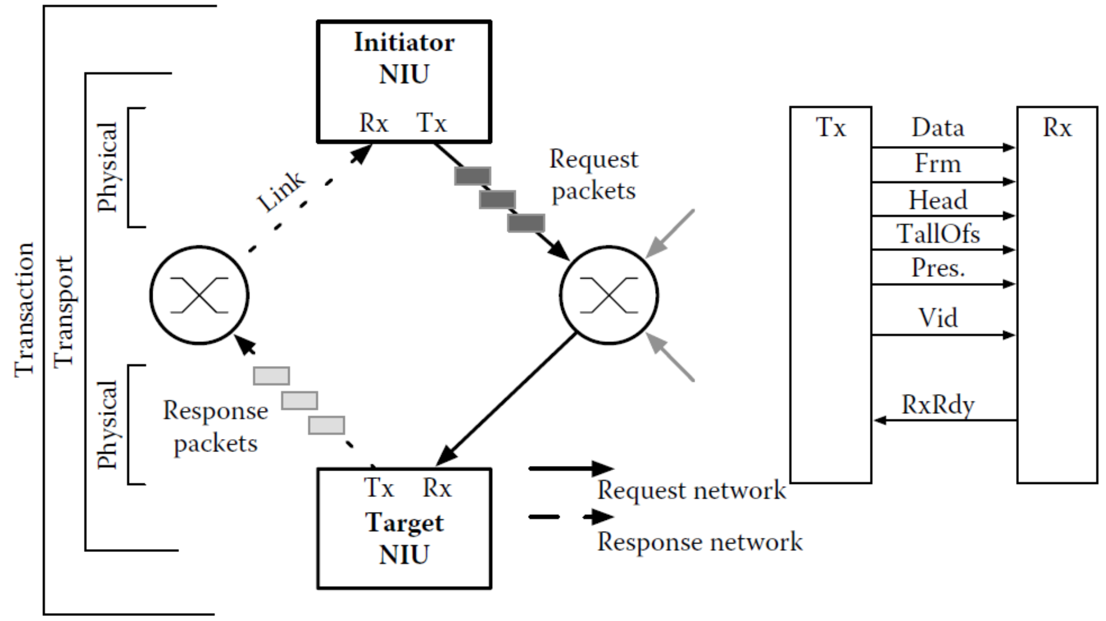
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port":  A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

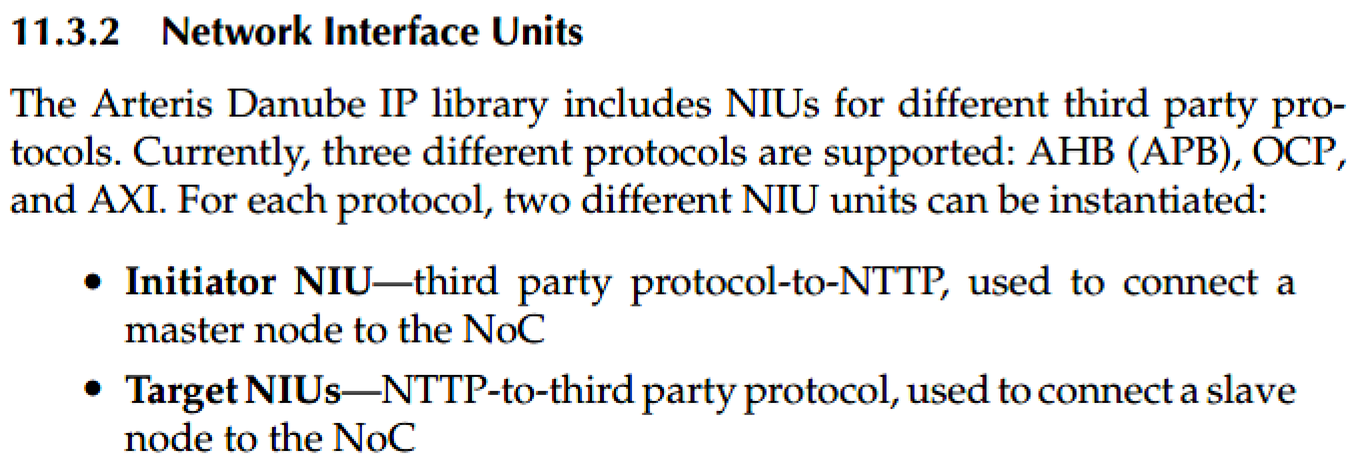
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper;	The Arteris NoC utilized by the Exynos SoC buffers by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4855-4585-1715.1

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

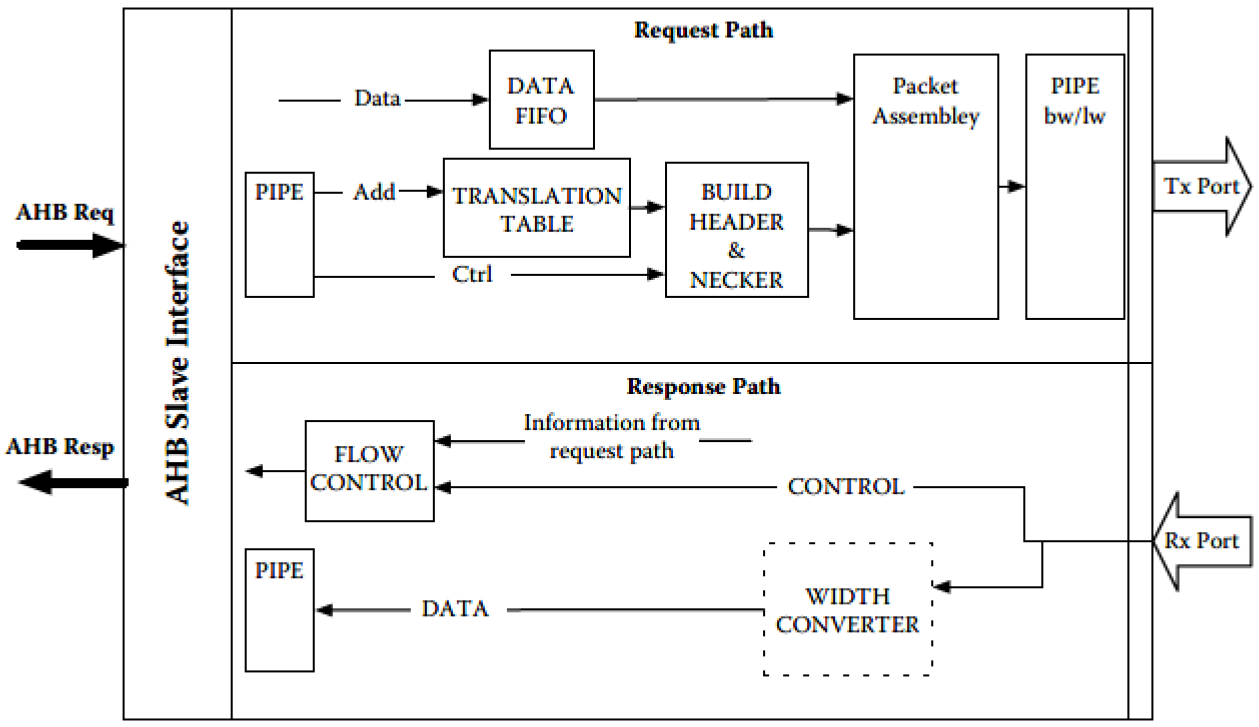
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

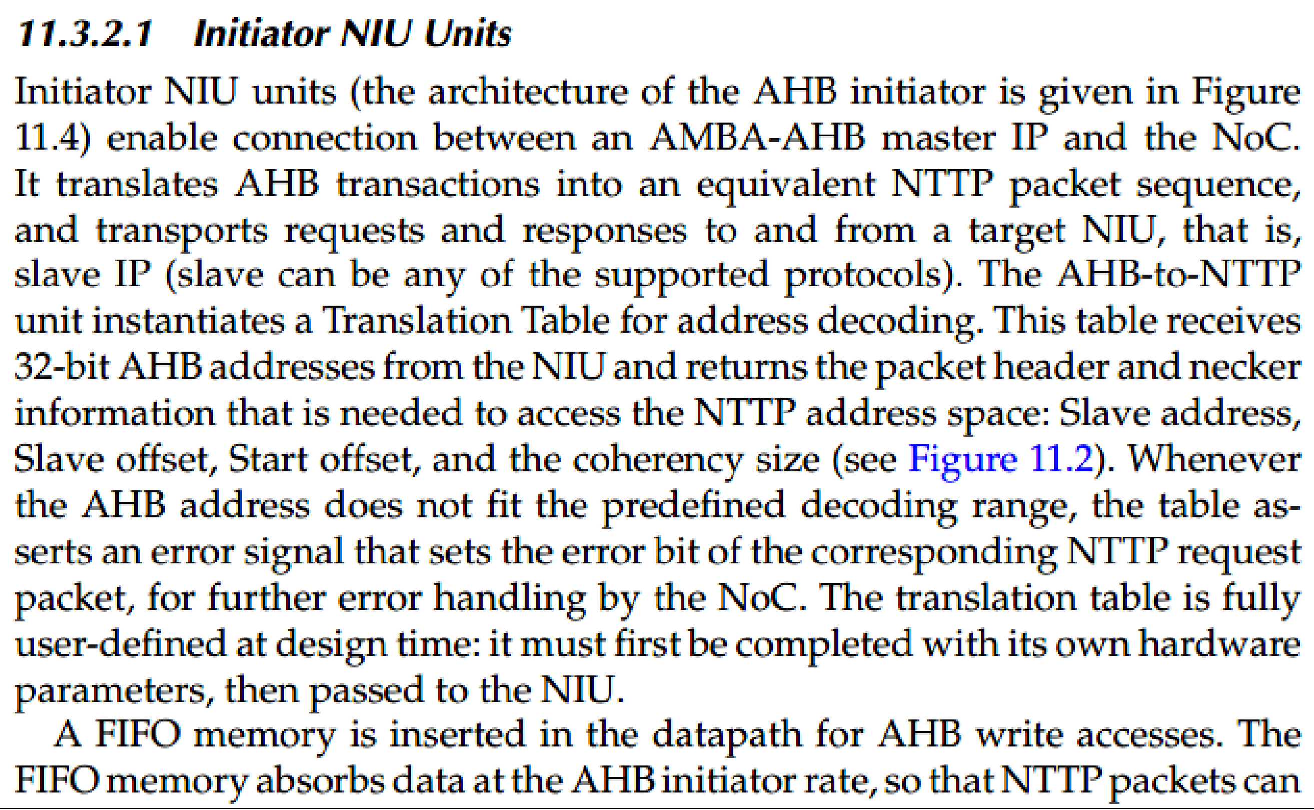
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

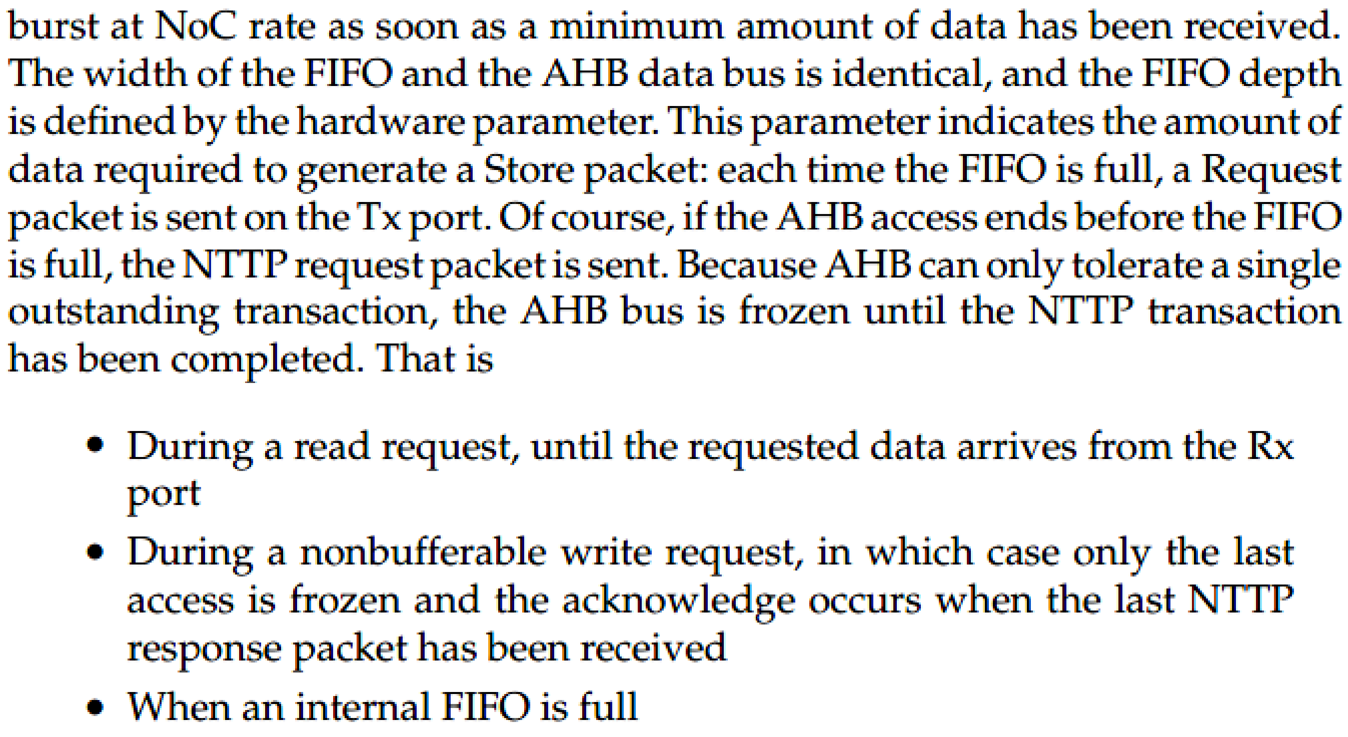
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

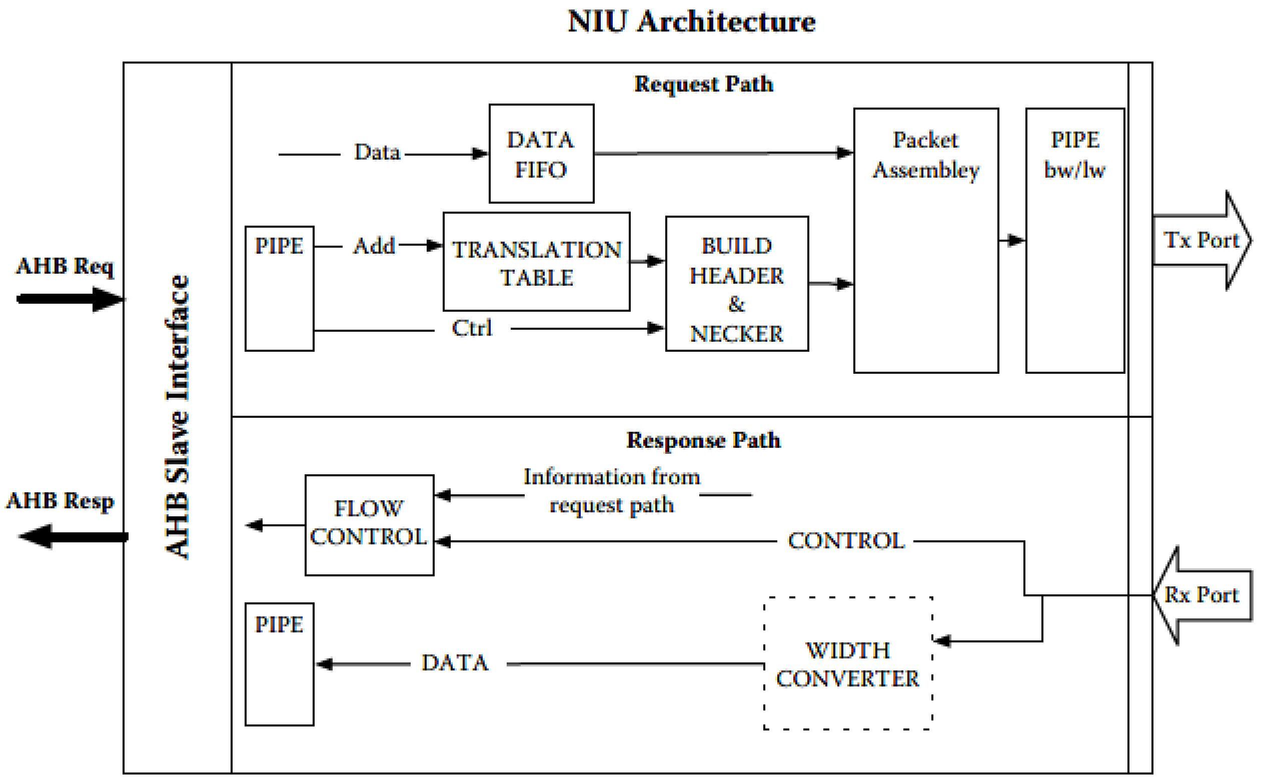
4855-4585-1715.1

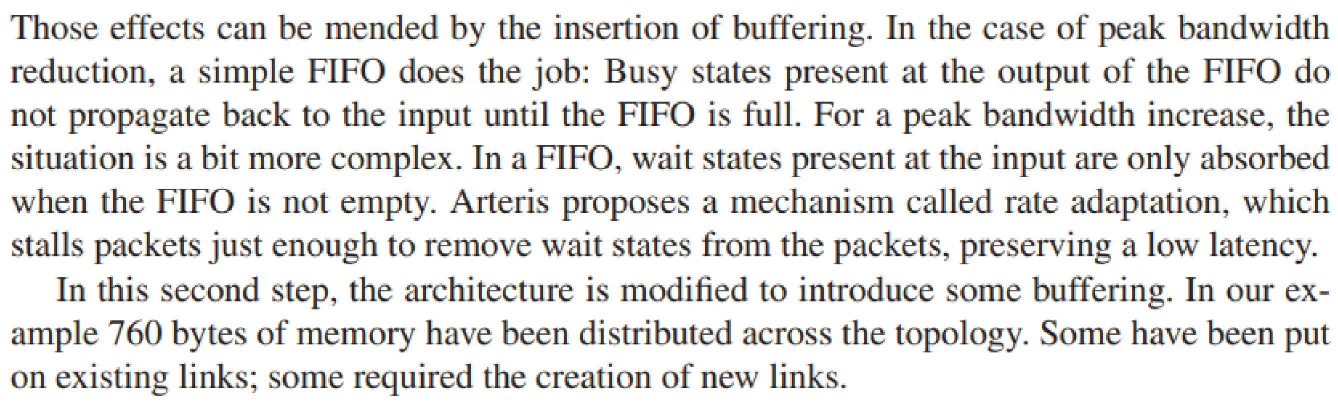
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

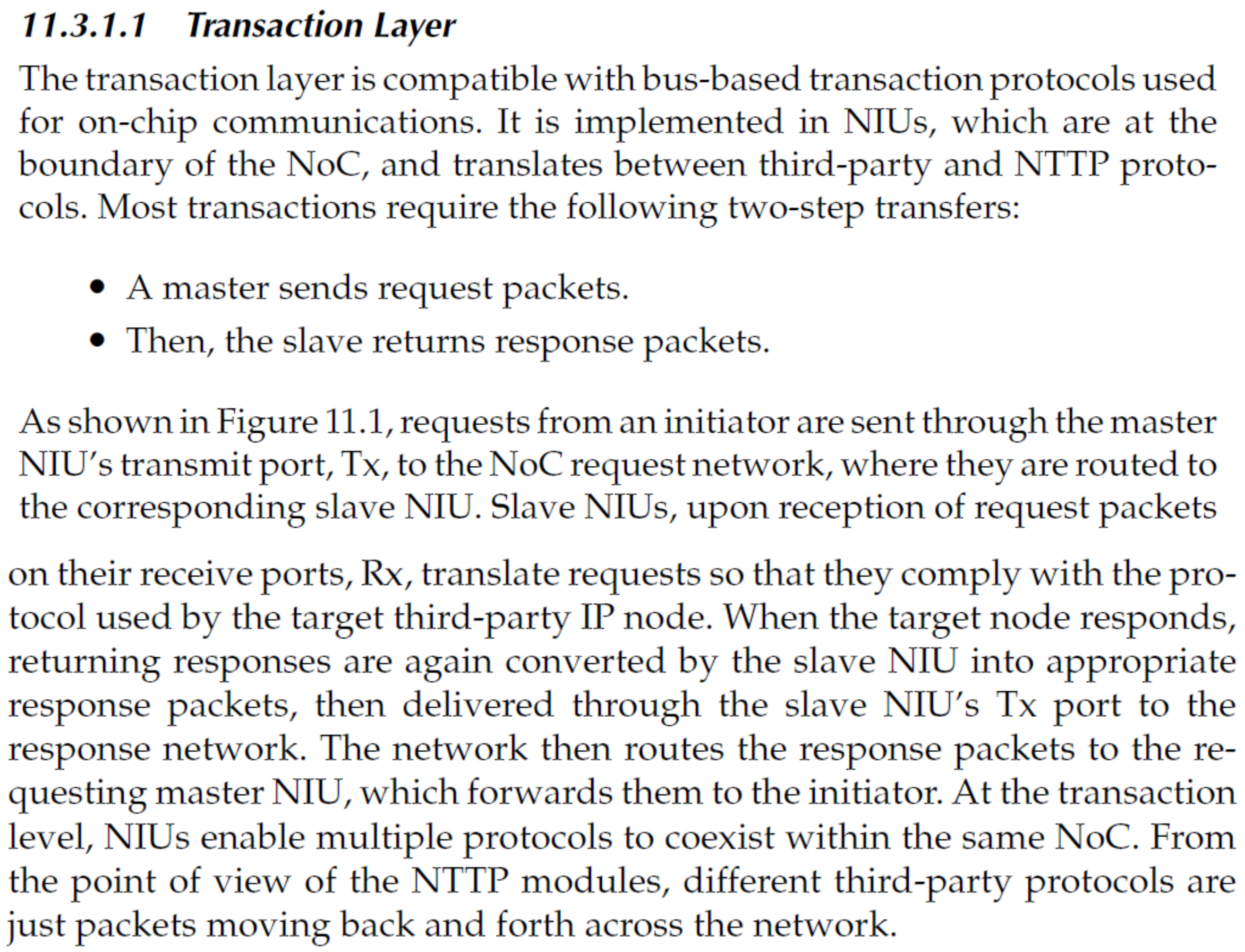
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	NIU Architecture  Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path Information from request path AHB Resp FLOW CONTROL CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port":  ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

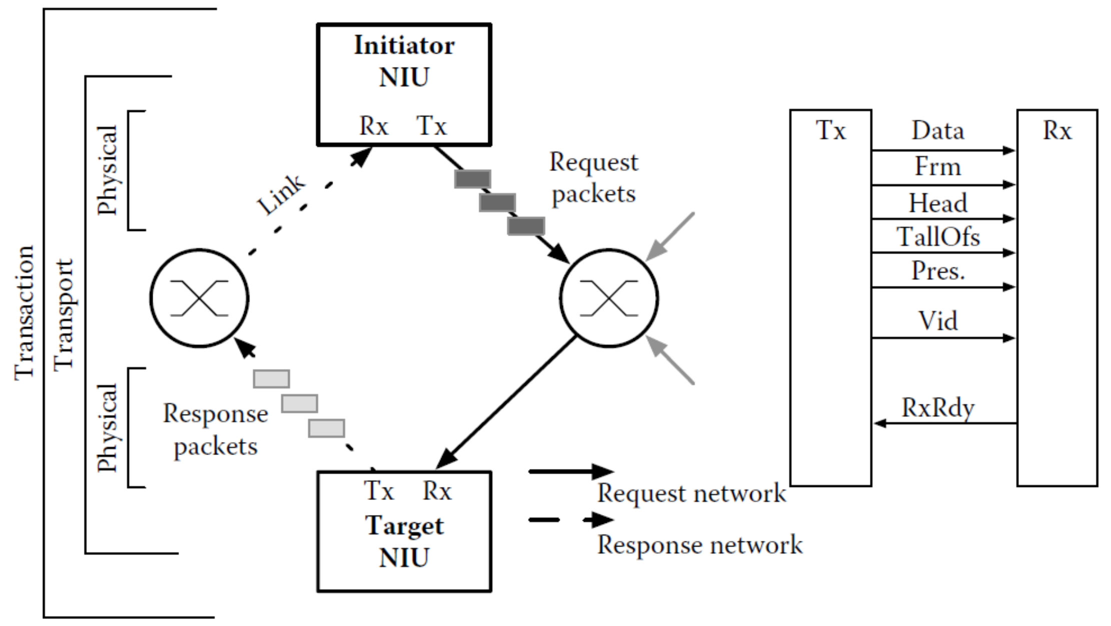
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

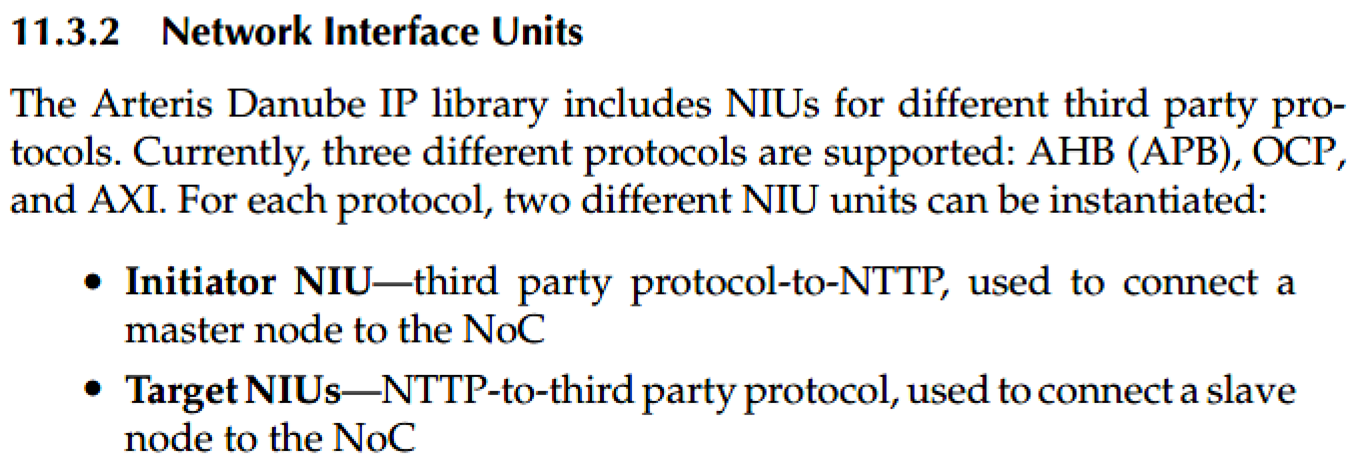
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface AHB Resp Response Path Information from request path FLOW CONTROL CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its

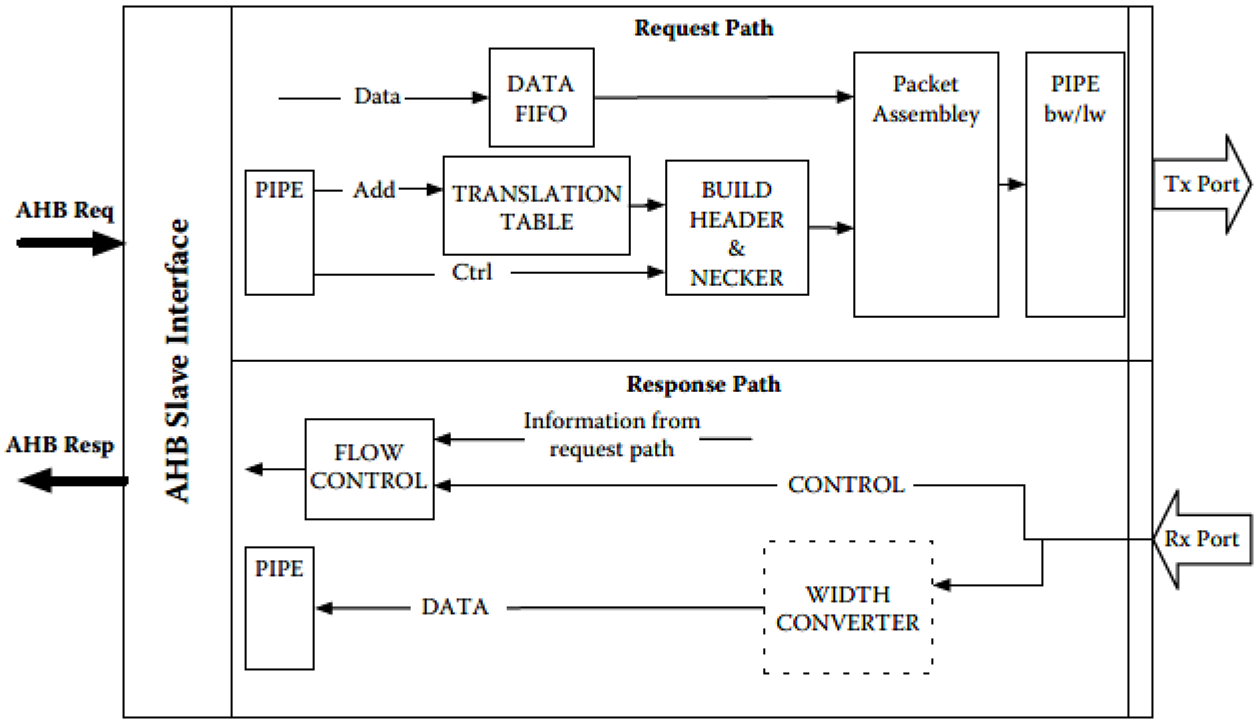
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level":  Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.
transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,	The Arteris NoC utilized by the Exynos SoC transfers the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

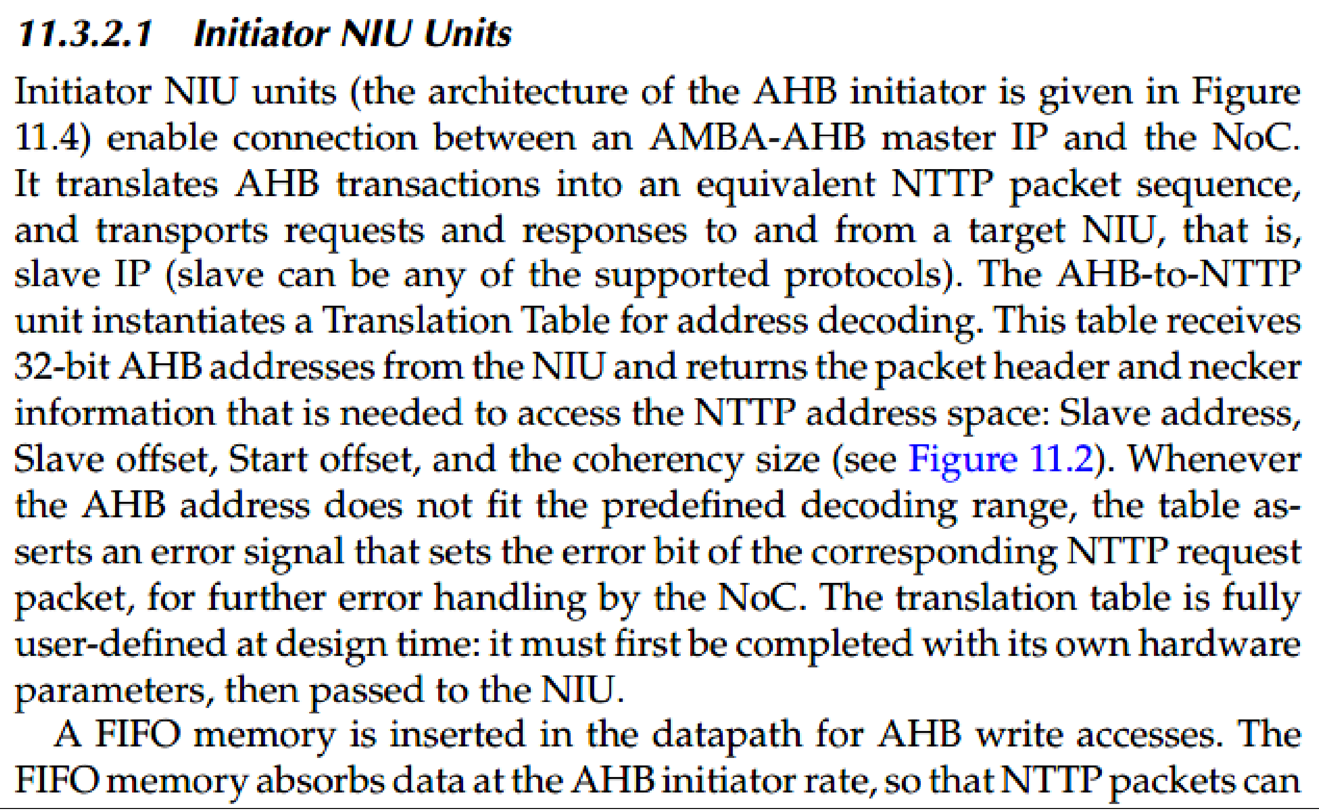
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Transaction Transport Physical Physical Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

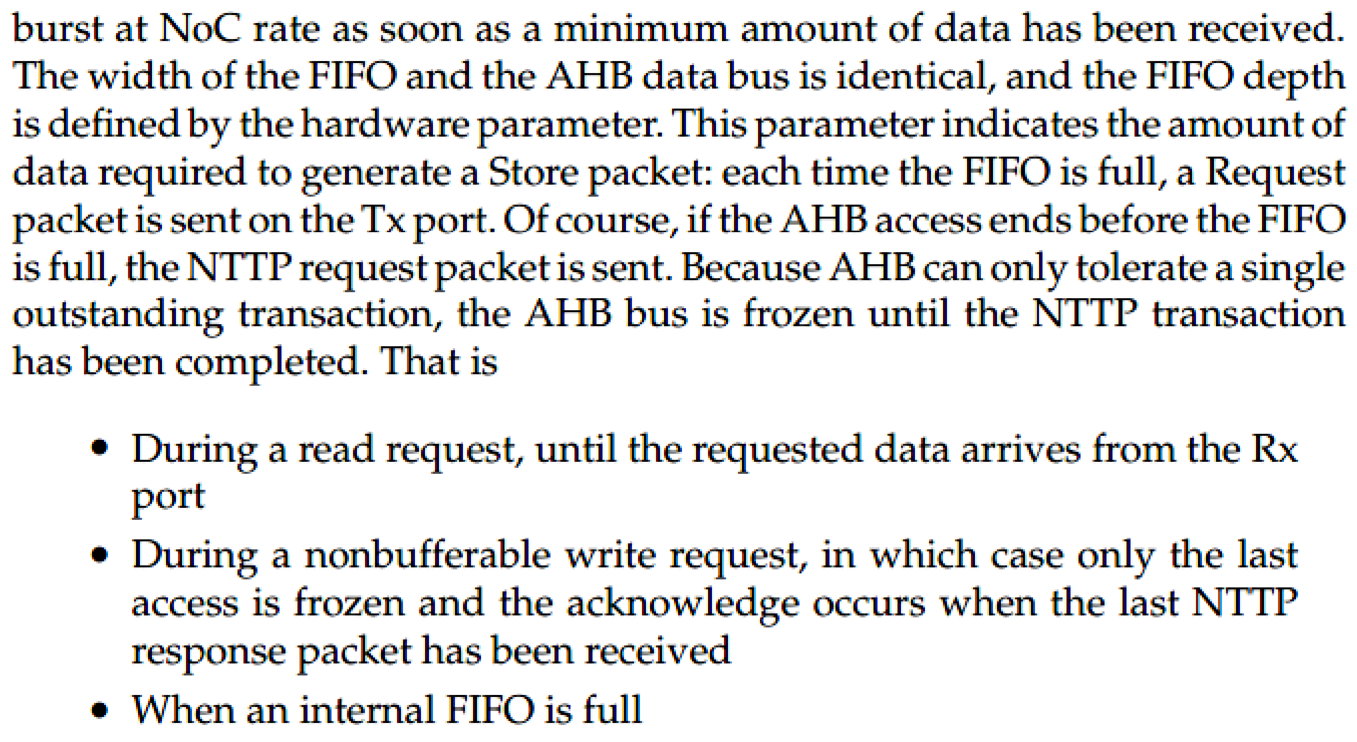
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	NIU Architecture  Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path Information from request path FLOW CONTROL CONTROL AHB Resp Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

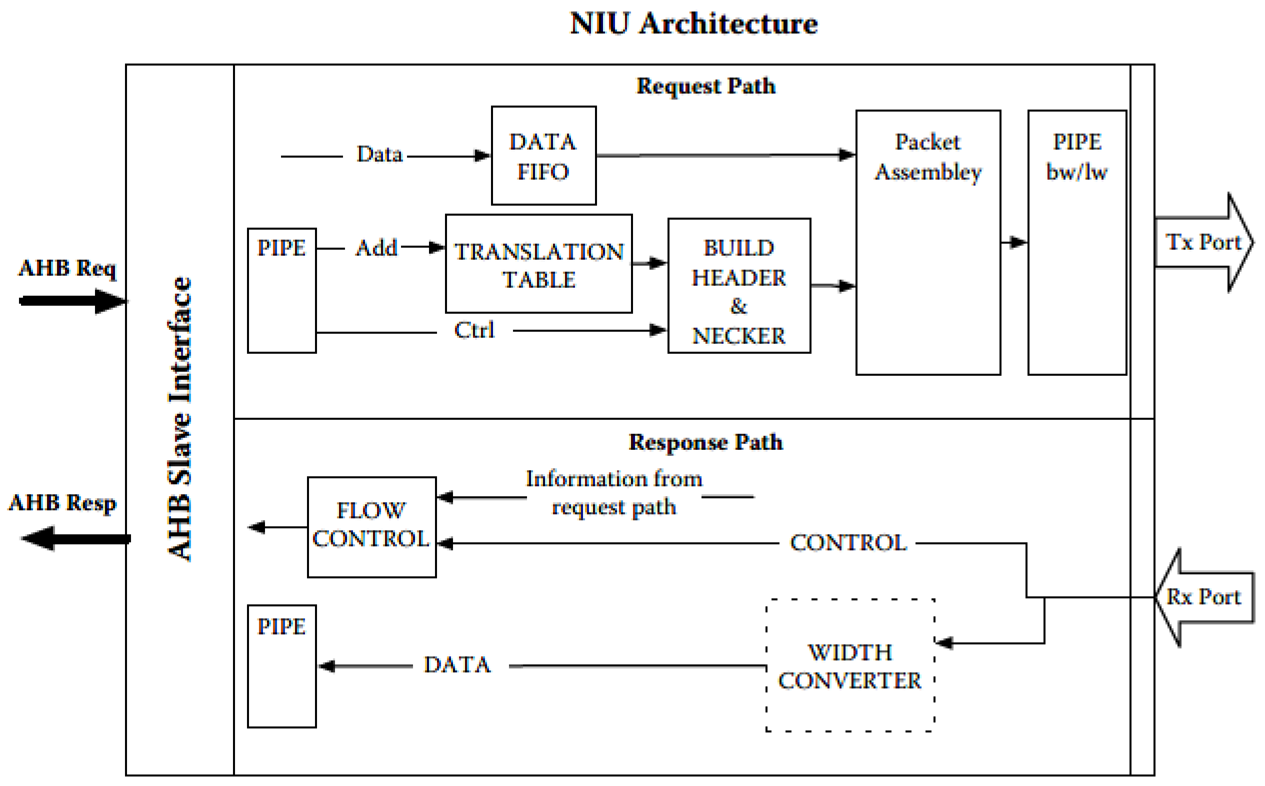
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port":  ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

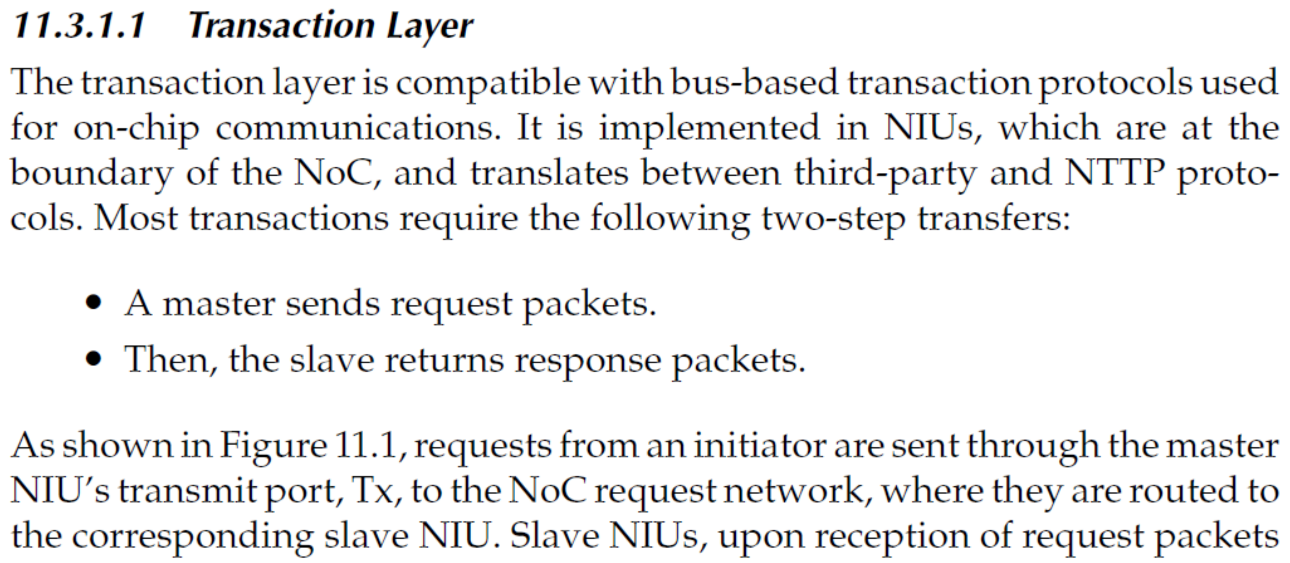
4855-4585-1715.1

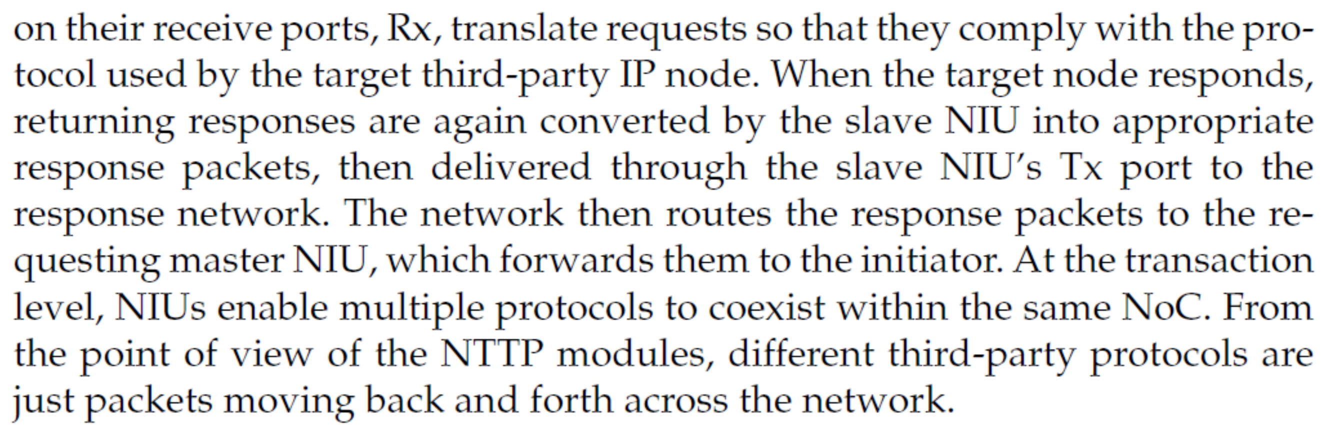
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

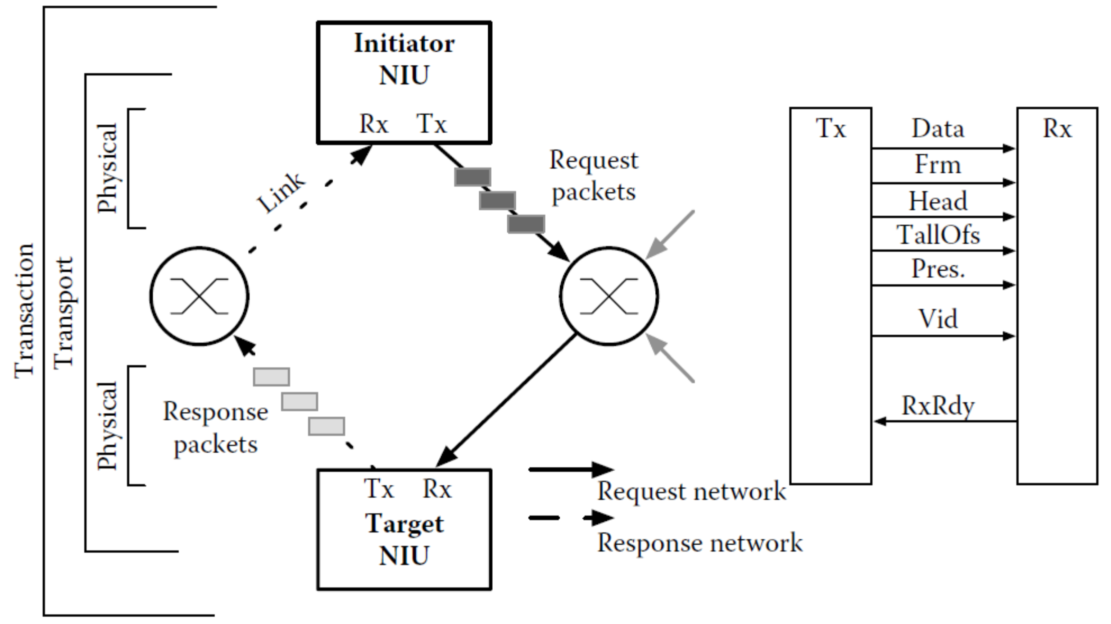
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path Information from request path FLOW CONTROL CONTROL AHB Resp Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
wherein at least one of the first determination unit	In the Arteris NoC utilized by the Exynos SoC, at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the

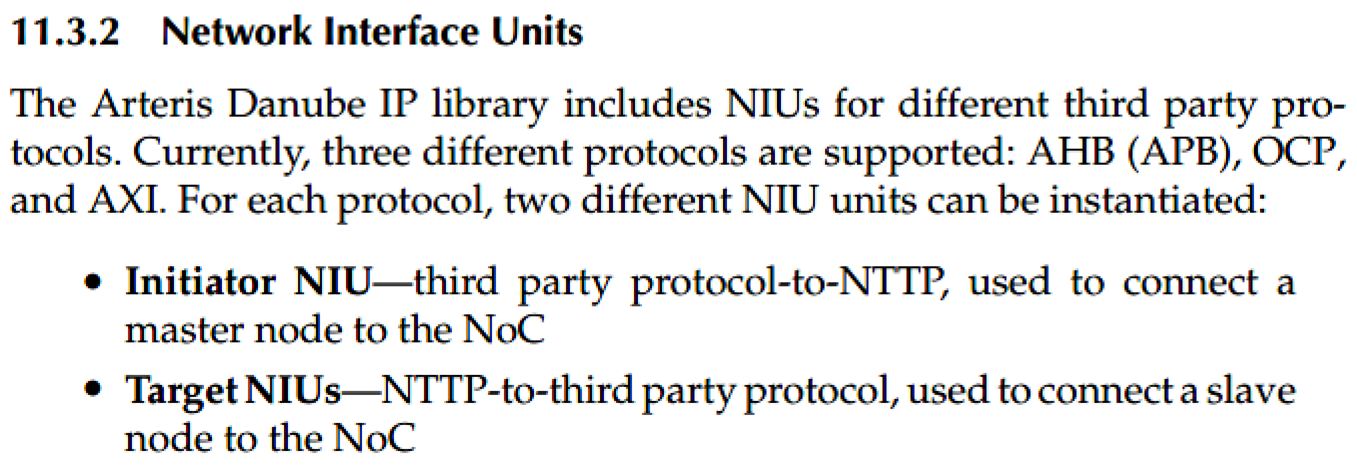
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it	communication between the master and the slave wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

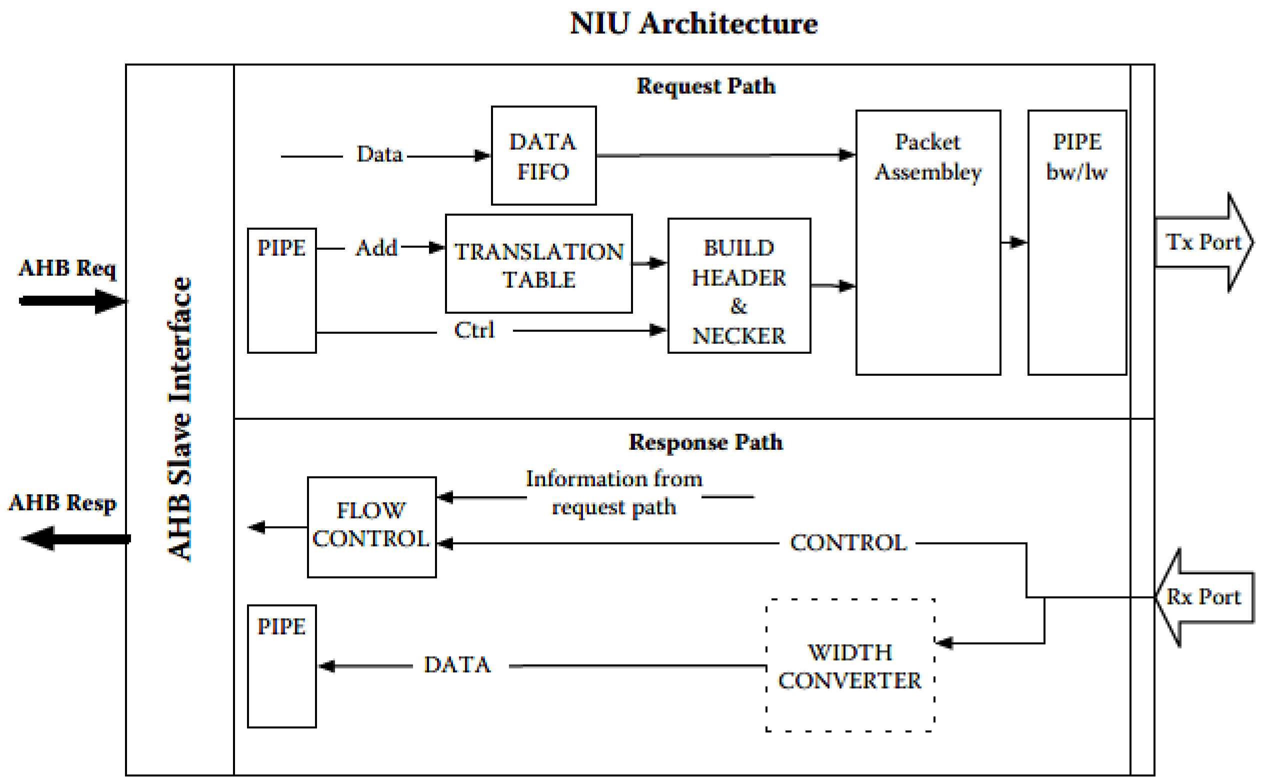
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

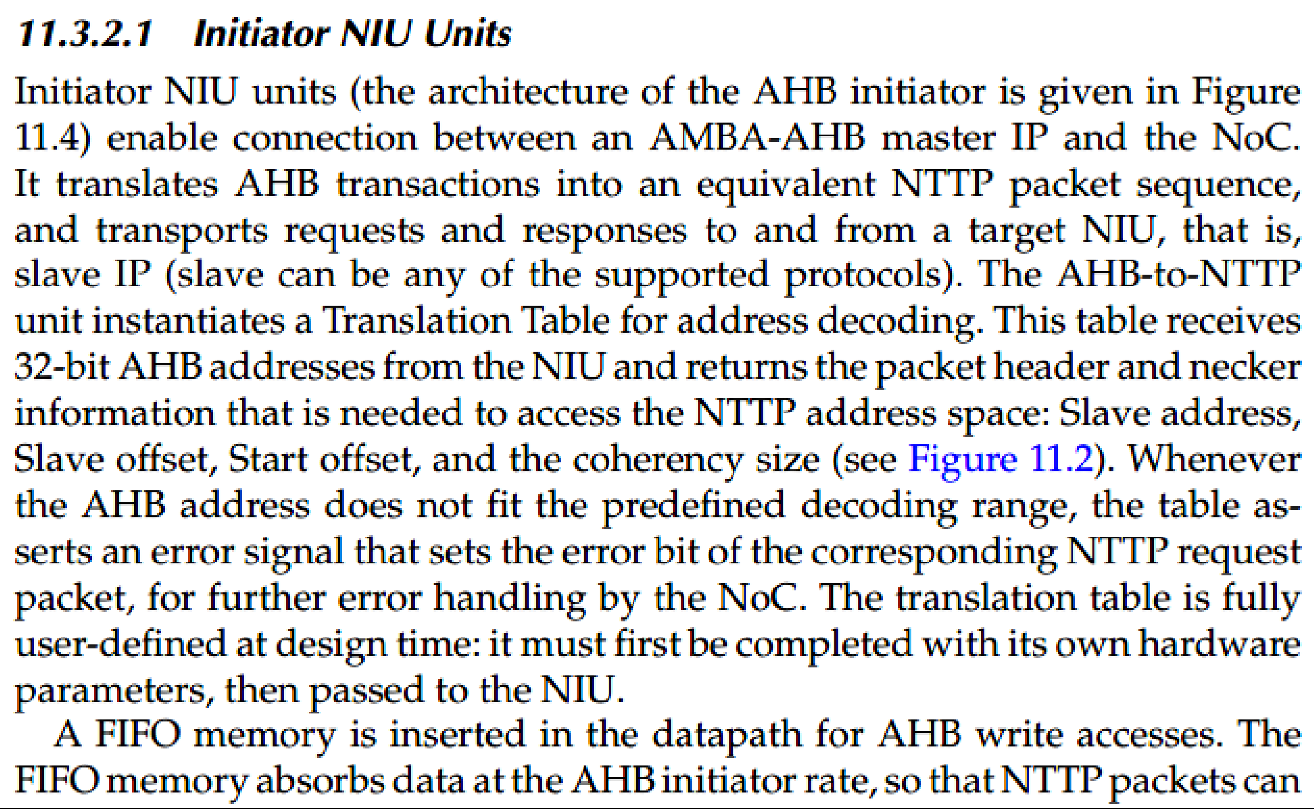
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

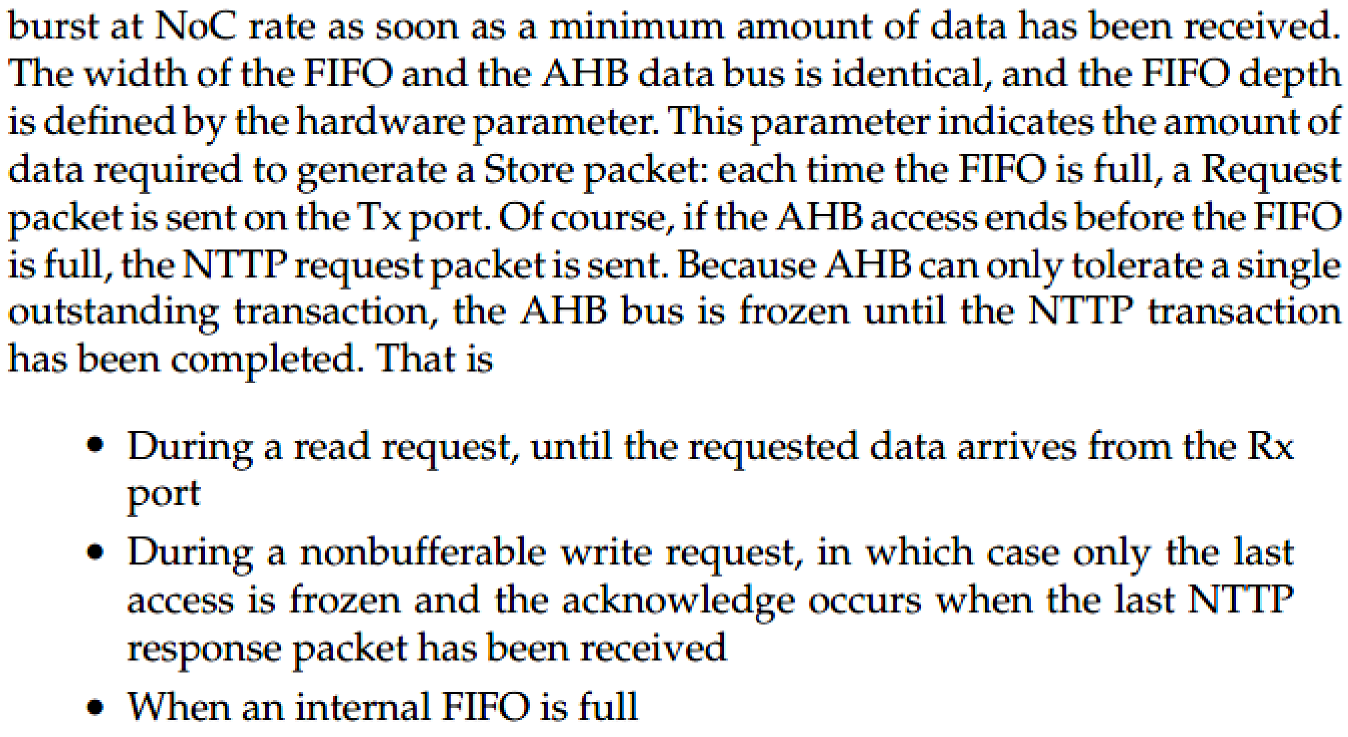
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Initiator NIU Rx Tx Link Request packets Transaction Transport Physical Physical Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC" and the Target NIUs are "used to connect a slave node to the NoC":

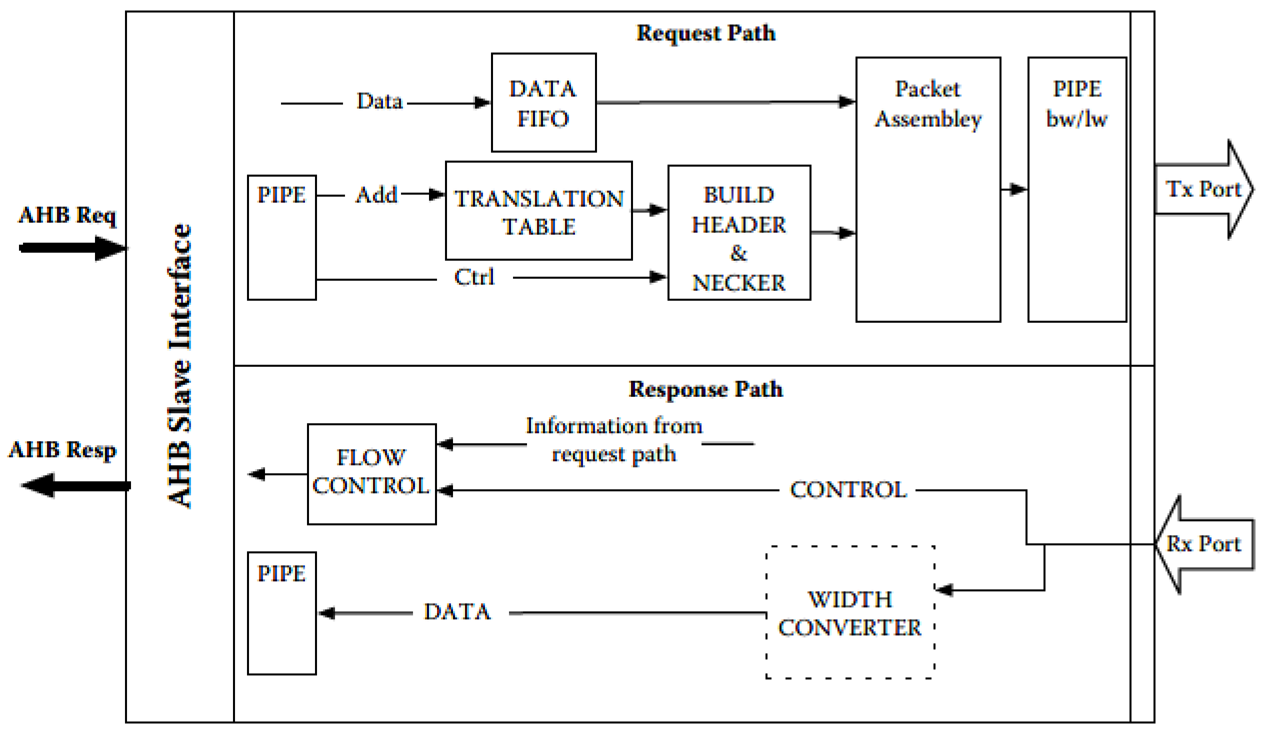
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 **11.3.2 Network Interface Units** The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

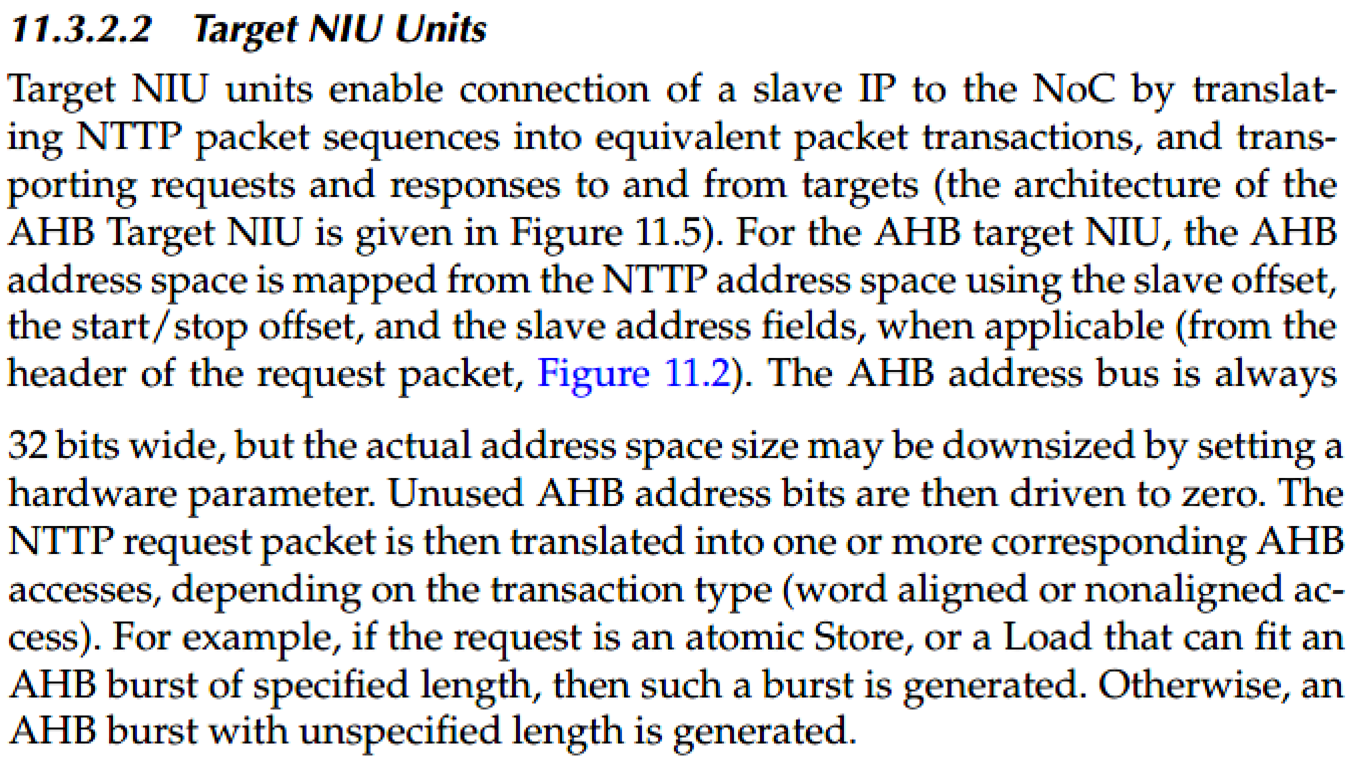
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

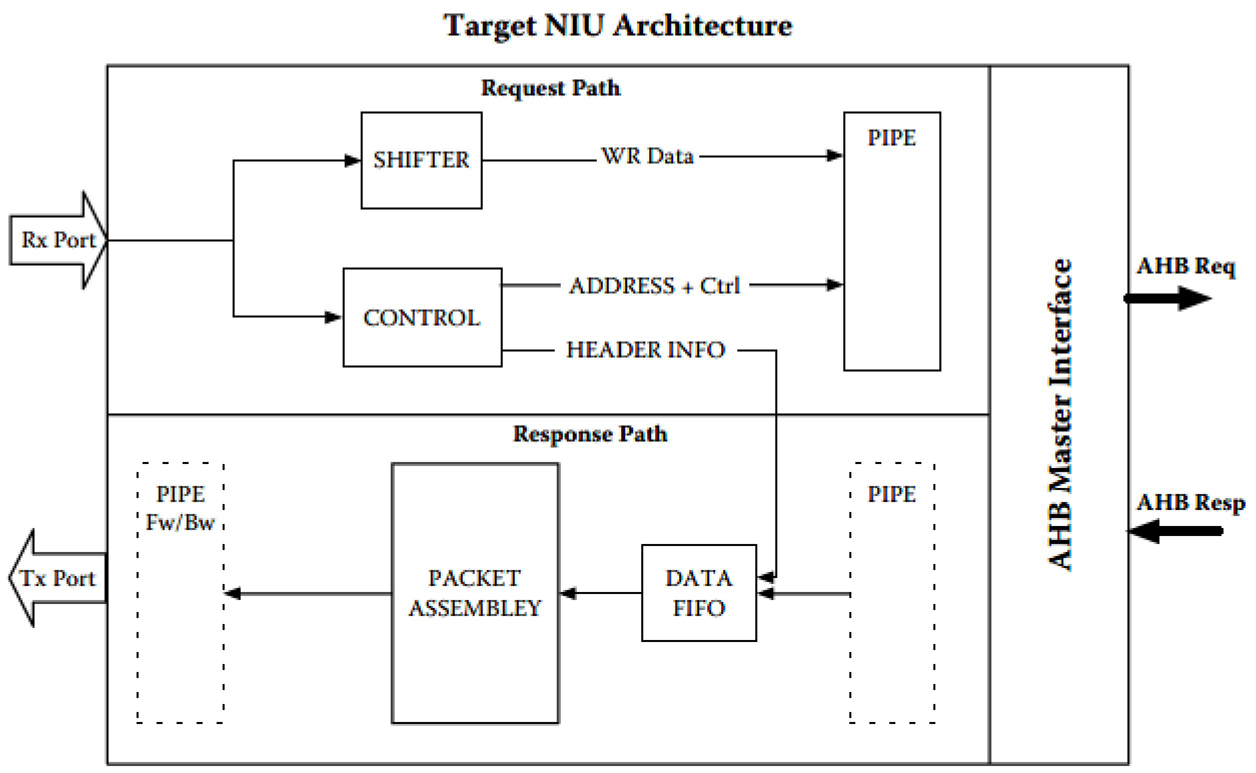
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 NIU Architecture Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl Tx Port AHB Req AHB Slave Interface Response Path AHB Resp Information from request path FLOW CONTROL CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

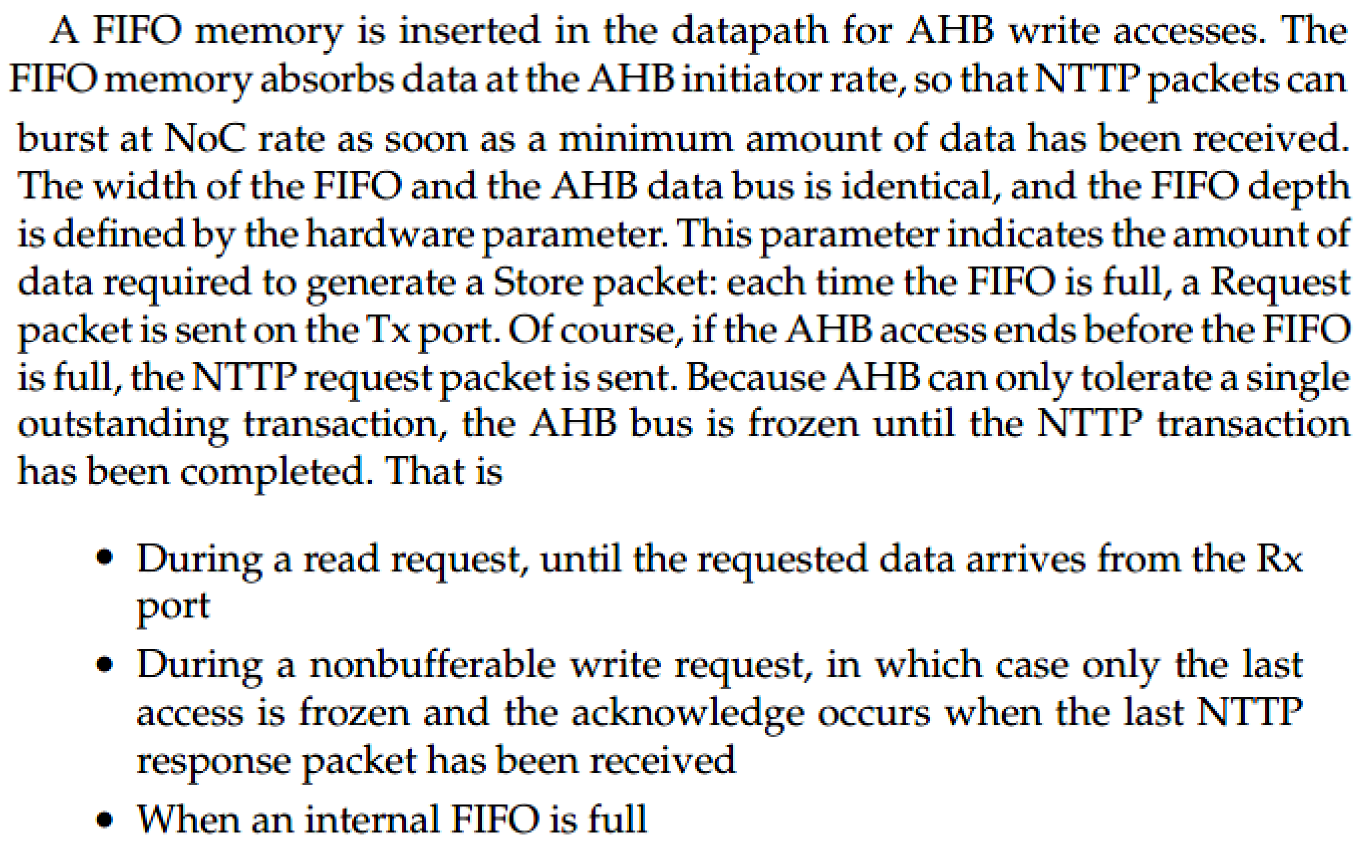
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port":  **_11.3.2.1 Initiator NIU Units_** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

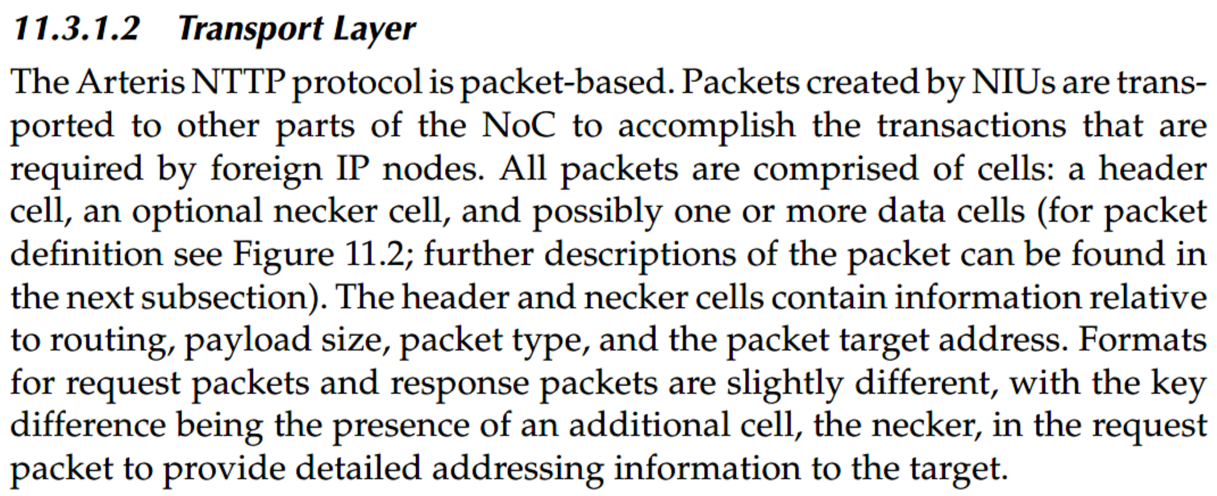
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

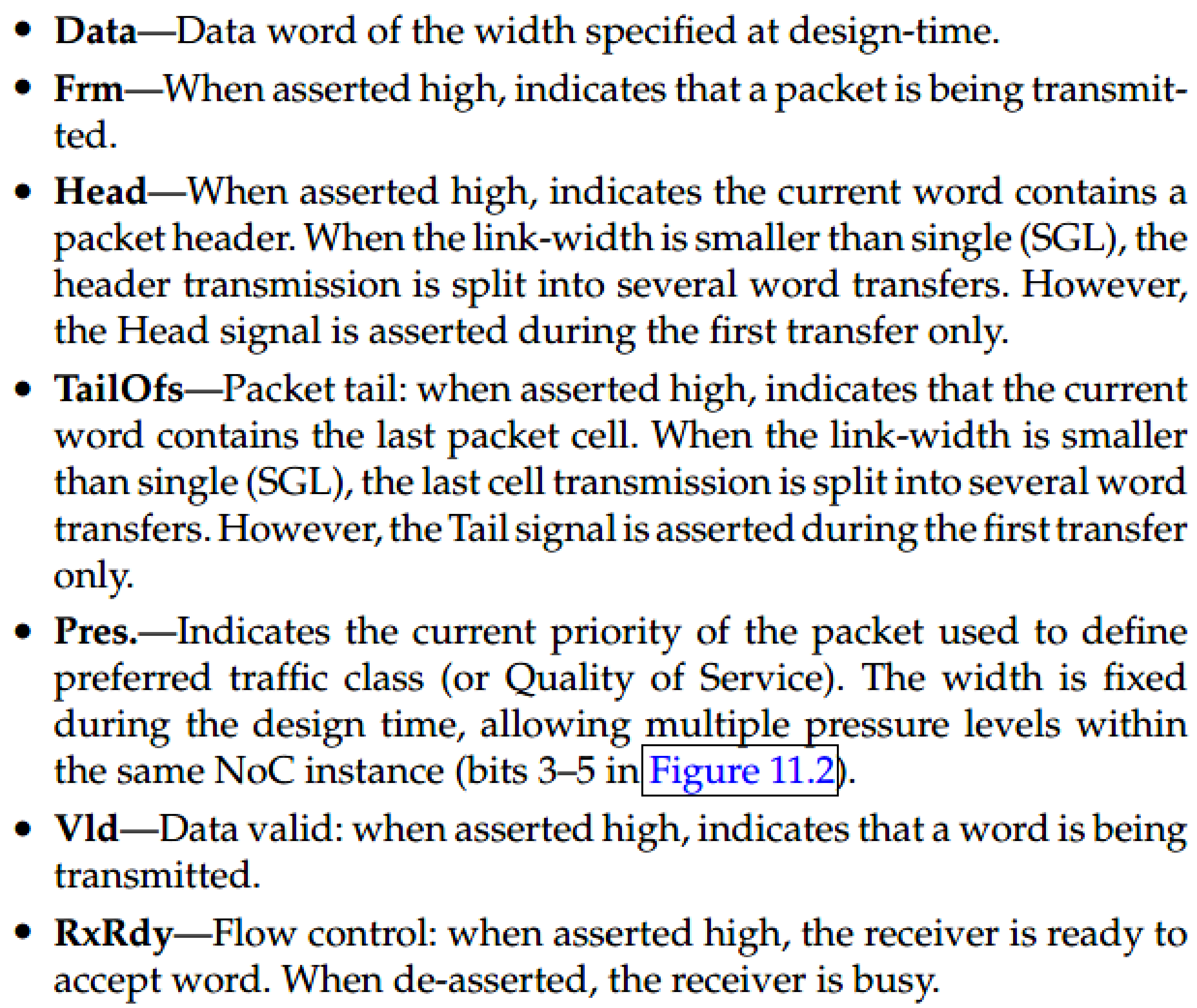
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	NIU Architecture  Request Path Data DATA FIFO Packet Assembley PIPE bw/lw PIPE Add TRANSLATION TABLE BUILD HEADER & NECKER Ctrl AHB Req AHB Slave Interface Tx Port Response Path AHB Resp FLOW CONTROL Information from request path CONTROL Rx Port PIPE DATA WIDTH CONVERTER **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

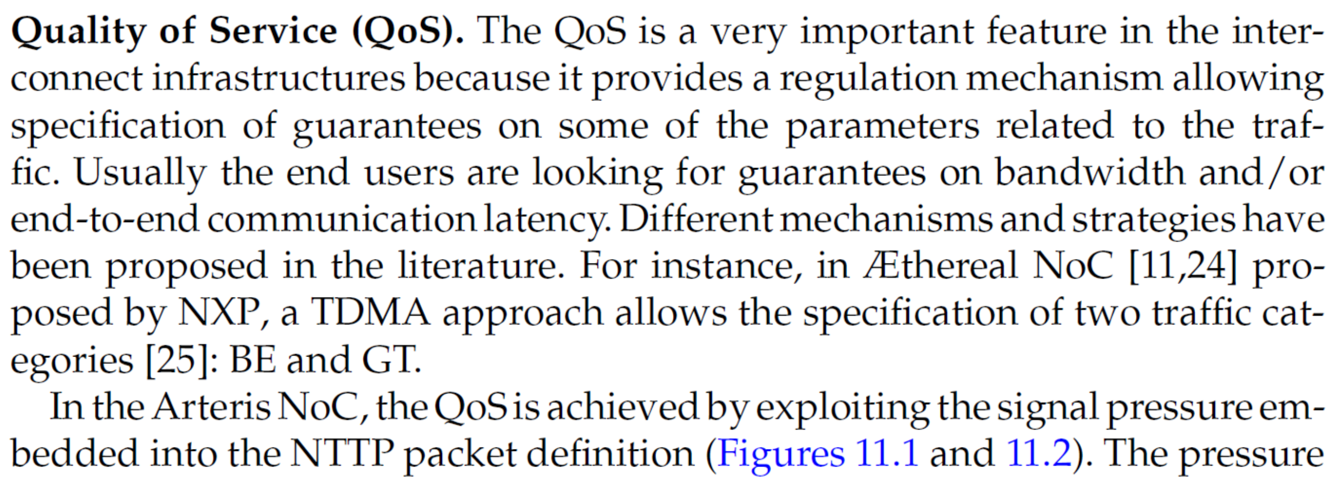
'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 Target NIU Architecture Request Path SHIFTER WR Data PIPE Rx Port CONTROL ADDRESS + Ctrl HEADER INFO AHB Master Interface AHB Req AHB Resp Response Path PIPE Fw/Bw PACKET ASSEMBLEY DATA FIFO PIPE Tx Port **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port":  A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":  ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As yet a further illustration, packets in the Arteris NoC are "delivered as words that are sent along links and "[o]ne link (represented in Figure 11.1) defines the following signals," which include "the current priority of the packet used to define preferred traffic class (or Quality of Service)" and "[f]low control":

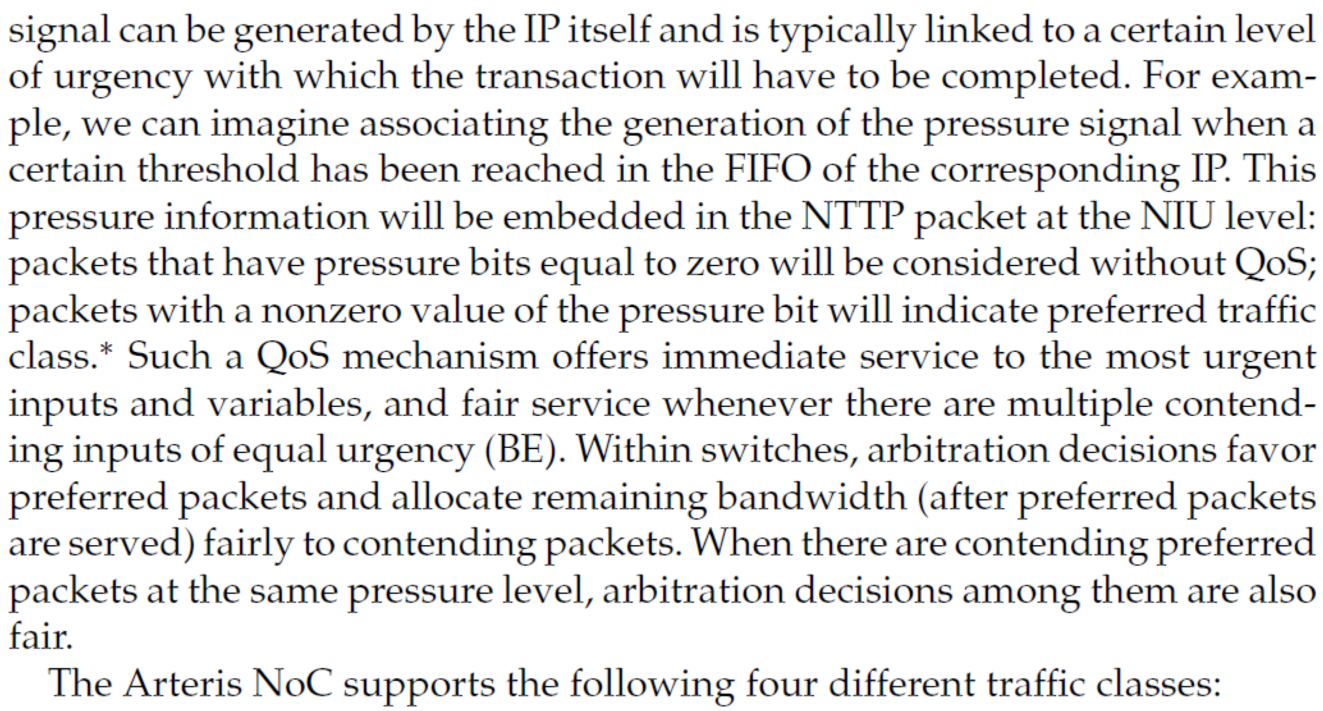
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 • **Data**—Data word of the width specified at design-time. • **Frm**—When asserted high, indicates that a packet is being transmitted. • **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only. • **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only. • **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2). • **Vld**—Data valid: when asserted high, indicates that a word is being transmitted. • **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy. *Id.* at 313-314.

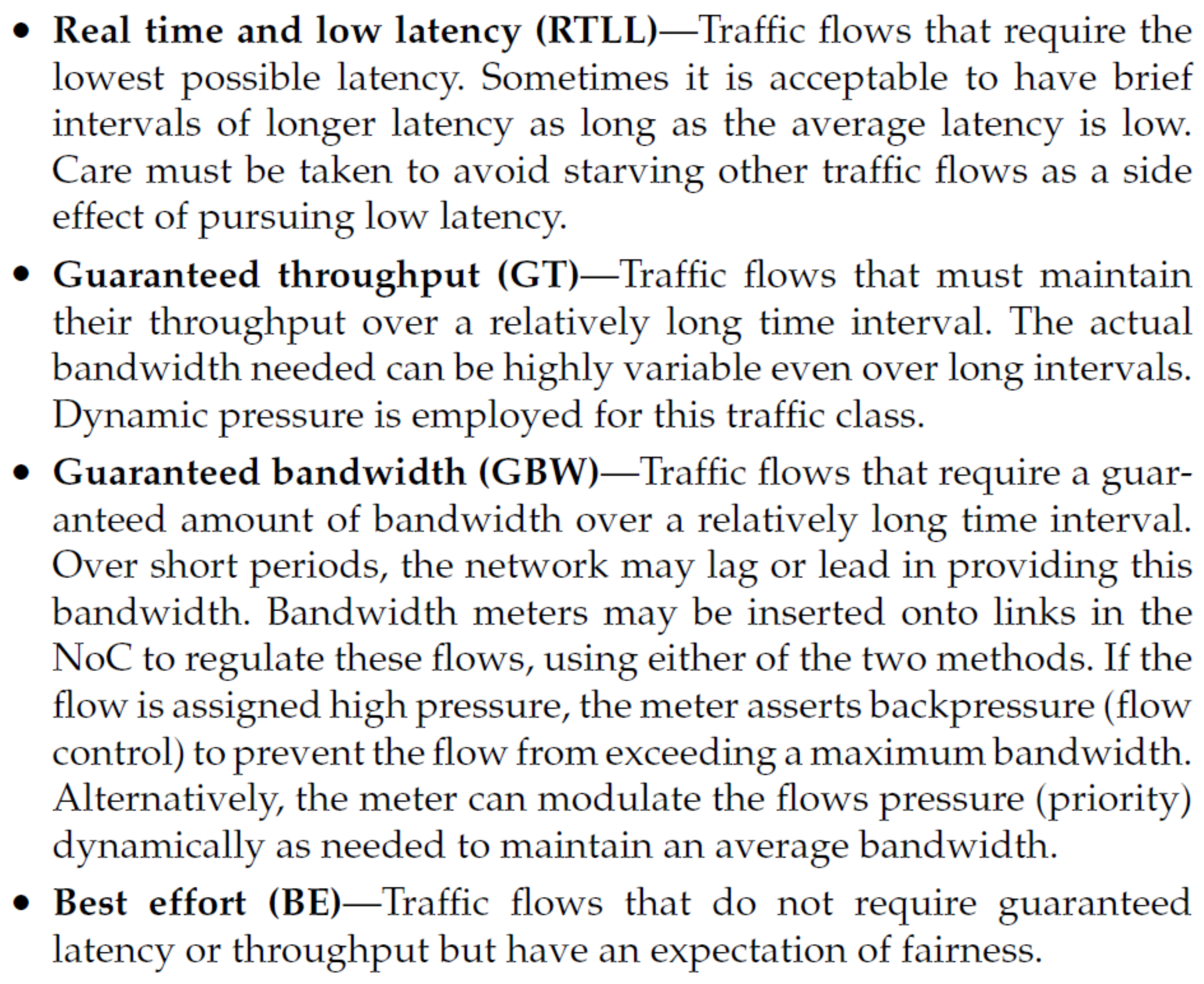
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	As a further illustration, the Arteris NoC implements Quality of Service (QoS) to "provide[] a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic"; "QoS, which includes guarantees of throughput and/or latency, is achieved by exploiting the signal pressure embedded into the NTTP packet definition" where the "pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed"; and the "pressure information will be embedded in the NTTP packet at the NIU level":  **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

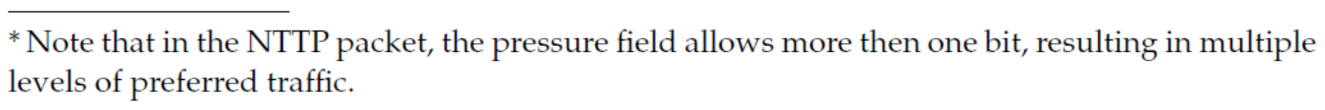
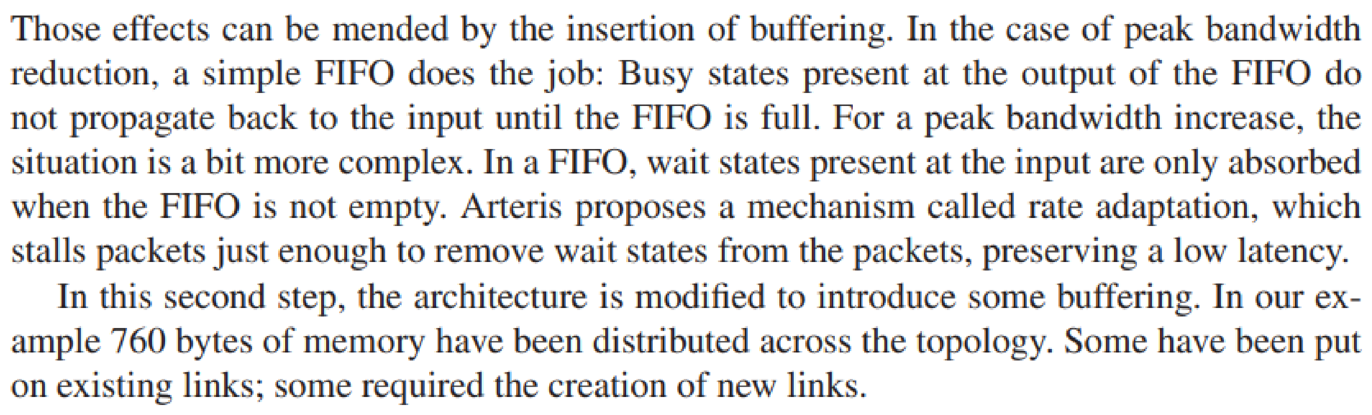
4855-4585-1715.1

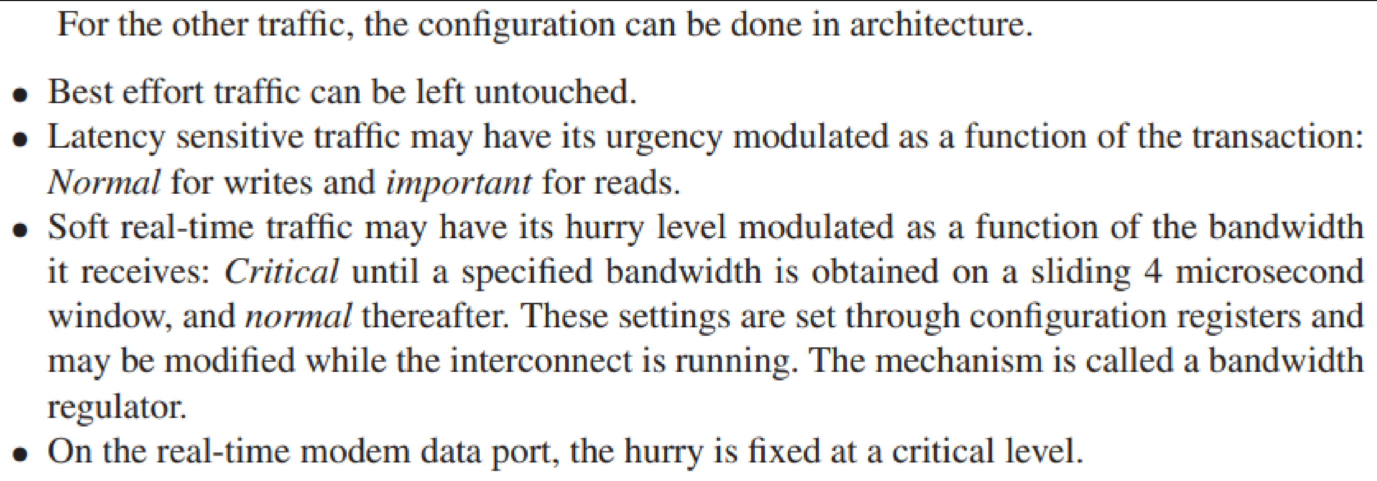
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

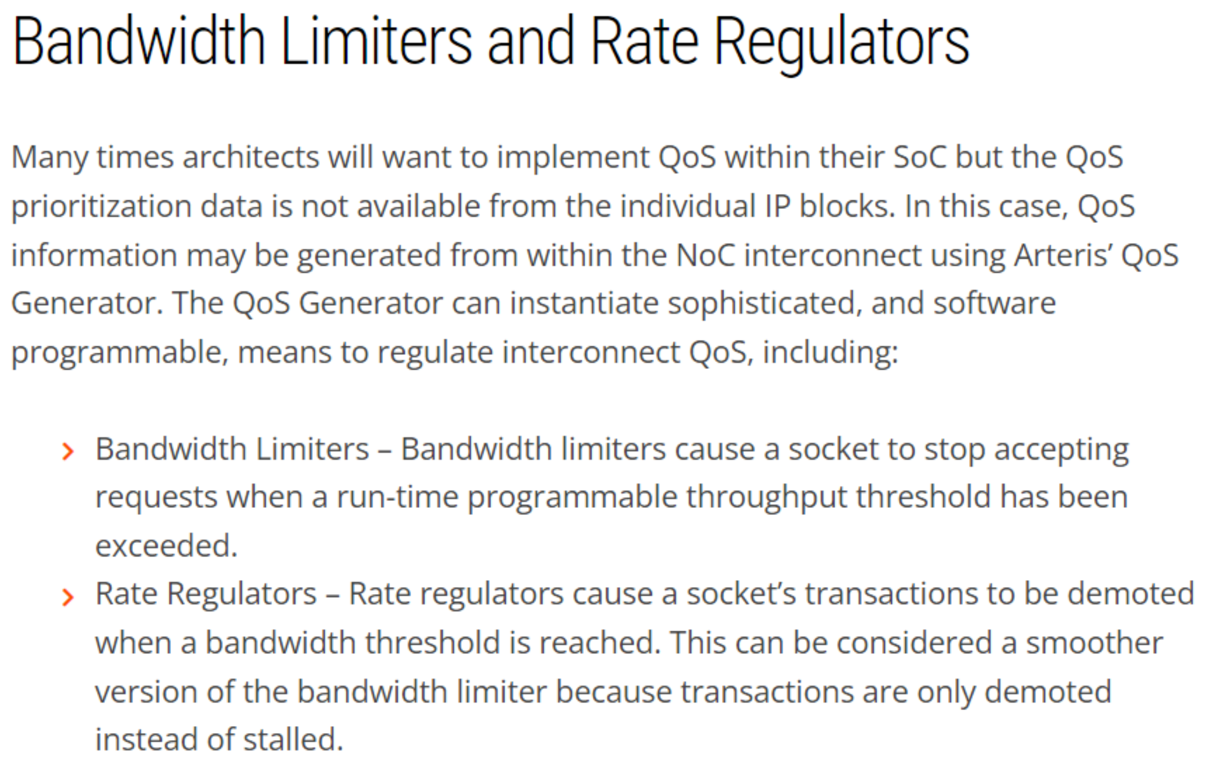
'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 • **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip [1]
	 * Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316. In addition, the Arteris Interconnect includes "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level."  Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springer-appdrivennocarchitecture8.5x11.pdf, at p. 16.

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 For the other traffic, the configuration can be done in architecture. • Best effort traffic can be left untouched. • Latency sensitive traffic may have its urgency modulated as a function of the transaction: *Normal* for writes and *important* for reads. • Soft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives: *Critical* until a specified bandwidth is obtained on a sliding 4 microsecond window, and *normal* thereafter. These settings are set through configuration registers and may be modified while the interconnect is running. The mechanism is called a bandwidth regulator. • On the real-time modem data port, the hurry is fixed at a critical level. *Id.* at 18. As a further illustration, the Arteris NoC implements QoS mechanisms that performs arbitration based on "Bandwidth Regulartor (BR)" and "Bandwidth Limiter (BL)":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Samsung Exynos 1280 System on Chip[1]
	 Bandwidth Limiters and Rate Regulators Many times architects will want to implement QoS within their SoC but the QoS prioritization data is not available from the individual IP blocks. In this case, QoS information may be generated from within the NoC interconnect using Arteris' QoS Generator. The QoS Generator can instantiate sophisticated, and software programmable, means to regulate interconnect QoS, including: › Bandwidth Limiters – Bandwidth limiters cause a socket to stop accepting requests when a run-time programmable throughput threshold has been exceeded. › Rate Regulators – Rate regulators cause a socket's transactions to be demoted when a bandwidth threshold is reached. This can be considered a smoother version of the bandwidth limiter because transactions are only demoted instead of stalled. https://www.arteris.com/end-to-end-quality-of-service-qos